

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, информатики и информационных технологий  
Кафедра информационно-коммуникационных технологий в образовании

# **РАЗРАБОТКА КАТАЛОГА ИНТЕРНЕТ-МАГАЗИНА**

*Выпускная квалификационная работа  
по специальности 09.03.022 – Информационные технологии в образовании*

Исполнитель: студент группы БС-4141z  
Института информатики и ИТ  
Казакбаев Т. У.

Руководитель: к.ф.н., доцент кафедры НИТО  
Грушевская В. Ю.

Работа допущена к защите  
« \_\_\_\_ » \_\_\_\_\_ 2016 г.  
Зав. кафедрой \_\_\_\_\_

Екатеринбург – 2016

## Реферат

Казакбаев Т.У. РАЗРАБОТКА КАТАЛОГА ИНТЕРНЕТ-МАГАЗИНА «УЧЕБНАЯ НАГРУЗКА» ДЛЯ УЧЕБНОЙ ЧАСТИ ВУЗА, выпускная квалификационная работа: 126 стр., рис. 109, библи. 30 + CD ДИСК

*Ключевые слова:* интернет-магазин, веб-сайт, CMS, система управления контентом

*Объект разработки* – ИНТЕРНЕТ-МАГАЗИН «ВИТЯЗЬ».

*Цель работы* – разработка интернет-магазина «Витязь», обеспечивающего все потребности интернет-маркетинга.

В работе приведены преимущества и новые возможности интернет-торговли. Рассматривается понятие интернет-магазина, структура интернет-магазина, средства и методы, используемые для его разработки. Рассмотрена CMS собственной разработки, описаны ее функции, средства и методы ее разработки. Дается описание интернет-магазина «Витязь», приведена структура данного сайта. Кроме того, в работе рассмотрен код и файлы, с помощью которых функционирует интернет-магазин.

<b>РЕФЕРАТ .....</b>	<b>2</b>
<b>ГЛАВА 1. ОПИСАНИЕ И ПРОЕКТИРОВАНИЕ ИНТЕРНЕТ-МАГАЗИНА. ....</b>	<b>7</b>
1.1 <b>АНАЛИЗ ИНТЕРНЕТ-ТОРГОВЛИ.....</b>	<b>7</b>
1.1.1 <i>Преимущества интернет-торговли.....</i>	<i>8</i>
1.1.2 <i>Новые возможности интернет-торговли.....</i>	<i>9</i>
1.2 <b>ПОНЯТИЕ ИНТЕРНЕТ-МАГАЗИНА .....</b>	<b>10</b>
1.3 <b>ТЕХНИЧЕСКОЕ ЗАДАНИЕ ИНТЕРНЕТ-МАГАЗИНА. ....</b>	<b>12</b>
1.3.1 <i>Используемые термины. ....</i>	<i>12</i>
1.3.2 <i>Выбор тематики интернет-магазина. ....</i>	<i>13</i>
1.3.3 <i>Структура интернет-магазина Витязь. ....</i>	<i>14</i>
1.3.4 <i>Метод разработки.....</i>	<i>35</i>
1.4 <b>ОПИСАНИЕ CMS СИСТЕМЫ.....</b>	<b>40</b>
1.4.1 <i>Веб-технологии CMS.....</i>	<i>40</i>
1.4.2 <i>Фреймворк CMS системы .....</i>	<i>43</i>
1.4.3 <i>Файловая структура CMS.....</i>	<i>45</i>
1.4.4 <i>Панель администрирования CMS.....</i>	<i>48</i>
1.4.5 <i>Заключение .....</i>	<i>63</i>
<b>ГЛАВА 2. РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА «ВИТЯЗЬ» .....</b>	<b>64</b>
2.1 <b>РАЗМЕЩЕНИЕ ФАЙЛОВ МОДУЛЕЙ.....</b>	<b>64</b>
2.2 <b>КОД МОДУЛЯ ДО МОДИФИКАЦИИ .....</b>	<b>65</b>
2.3 <b>API CMS СИСТЕМЫ.....</b>	<b>66</b>
2.4 <b>МОДУЛИ ИНТЕРНЕТ-МАГАЗИНА – ВИТЯЗЬ .....</b>	<b>71</b>
2.4.1 <i>Базовый контроллер– Main.php.....</i>	<i>72</i>
2.4.2 <i>Контроллер каталога - Kateg.php.....</i>	<i>80</i>
2.4.3 <i>Контроллер карточки товара – Tovari.php.....</i>	<i>82</i>
2.4.4 <i>Контроллер текстовой страницы – Cont.php .....</i>	<i>90</i>
2.4.5 <i>Контроллер новостей – Novosti.php .....</i>	<i>91</i>
2.4.6 <i>Контроллер страницы контактов – Contacts.php.....</i>	<i>95</i>
2.4.7 <i>Контроллер для работы с пользователями – Users.php.....</i>	<i>96</i>
2.4.8 <i>Контроллер обработки заказов – Zakazi.php.....</i>	<i>103</i>
2.4.9 <i>Контроллер обработок заявок с сайта – Zakzvon.php.....</i>	<i>110</i>
2.4.10 <i>Контроллер корзины – Korzina.php .....</i>	<i>111</i>
2.4.11 <i>Контроллер отзывов – Otzivi.php .....</i>	<i>120</i>

<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>123</b>
<b>ЛИТЕРАТУРА .....</b>	<b>124</b>

## Введение

Интернет развивается, создаётся огромное множество веб-приложений, предоставляются разнообразные услуги, продаются товары, обрабатываются файлы. В настоящее время в Интернете можно найти практически любой онлайн-сервис под свои нужды.

Онлайн-сервис – это сайт, который позволяет нам быстро и эффективно решить свою задачу. Работая в онлайн-сервисе, нам не надо устанавливать программы на свой компьютер, отсутствуют проблемы в совместимости между той или иной программой и операционной системой на компьютере пользователя, нет необходимости отслеживать обновления, онлайн-сервис — это всегда актуальная версия программы, доступен всегда и везде, он не привязан к конкретному компьютеру. Все что нужно для работы — это подключение к Интернету, все действия можно осуществлять через браузер.

Несмотря на многообразие онлайн-сервисов, одним из самых актуальных в сфере интернет-торговли, является интернет-магазин. Интернет-магазин очень удобен тем, что он работает круглые сутки и без выходных, посмотреть каталоги, приобрести товар, может покупатель из любой точки мира где есть доступ к интернету, в отличие от обычных магазинов, в которых площадь помещения ограничена, на сайтах можно разместить сколько угодно товара, рекламы и информации.

Целью данной выпускной квалификационной работы является, разработка полноценного интернет-магазина с современным интерфейсом, в котором будут:

- Каталог товаров
- Корзина покупателя
- Новости
- Отзывы пользователей
- Личный кабинет

*Объект разработки* – интернет-магазин «Витязь».

Интернет-магазин будет разработан на CMS<sup>1</sup> системе собственной разработки, в которой администратору сайта будут доступны возможности изменения данных на сайте, создания новых разделов, модерация<sup>2</sup> заказов и отзывов пользователей сайта. Дизайн и HTML верстка интернет-магазина будет предоставлена веб-студией.

Для достижения поставленной цели выпускной квалификационной работы необходимо решить следующий ряд задач:

- Описать, что такое интернет-торговля, интернет-магазин, почему актуально иметь интернет-магазин, плюсы и минусы интернет-магазина, что такое сайт, что такое каталог, что такое CMS система.
- Описать структуру создаваемого интернет-магазина, использованные технологии и языки программирования.
- Изучить структуру и документацию CMS системы.
- Развернуть CMS систему на веб-сервере.
- Разработать интернет-магазин на CMS.

---

1.CMS - (content management system) система управления контентом.

2.Модерация — осуществление контроля, за информацией, размещаемой пользователями на интернет-сайтах.

# **Глава 1. Описание и проектирование интернет-магазина.**

## **1.1 Анализ интернет-торговли**

С конца 90-х годов прошлого века, с появлением новых информационных технологий, динамичным развитием сети Интернет и повышением числа ее пользователей, приводя к новым областям исследования, как в мировой, так и отечественной экономики. В условиях единого информационного пространства, которое объединяет огромное число предприятий, фирм и государственных учреждений по всему миру, появляются новые условия для развития бизнеса. Развитие технологий в настоящее время приводит к формированию новых рынков, стимулирует спрос на принципиально новые товары и услуги, позволяют совершать сделки удаленно в режиме on-line (реального времени), изменяет традиционную производственно-коммерческую деятельность.

С начала 2000-го года Интернет становится не только средой общения и местом для поиска информации, но и площадкой для выполнения сделок по купле-продаже как электронного контента, так и в вещественной форме. Все это приводит к формированию в интернете действующих коммерческих организаций и открывает новые возможности для субъектов экономики (как для покупателей, так для продавцов). Наступает эра интернет-торговли.

Интернет-торговля — это реализация товаров или услуг через интернет-сайты. В режиме on-line, клиенты оформляют покупки, далее они выбирают вариант оплаты и доставки. Что позволяет клиенту доступно и удобно, совершать покупки, находясь дома или на работе.

На сегодняшний день интернет-торговля является активно развивающейся сферой предпринимательской деятельности. Этот вид бизнеса формирует новый виртуальный канал сбыта, который существенно сокращает

торговые расходы. С помощью интернет-торговли многие виды продукции могут быть доставлены в цифровой форме через интернет, а процесс оплаты, может быть осуществлен в режиме on-line. Еще несколько лет назад опыт покупок через интернет имели лишь немногие, а количество компаний, предлагающих свои товары и услуги в интернете, было крайне незначительным. В настоящее время количество on-line операций растет бурными темпами, а большинство компаний, активно начинают осваивать и использовать в своих целях преимущества интернет-торговли.

### **1.1.1 Преимущества интернет-торговли**

Интернет-торговля имеет ряд преимуществ:

1. Предоставляет глобальный доступ на глобальные рынки. Компания может расширить ассортимент товаров, а также свою базу клиентов.
2. Дает возможность улучшить бизнес-контакты. Покупатели могут наладить более тесные связи с продавцами товаров промышленного назначения.
3. В интернет-магазинах доступна информация о товарах и услугах в режиме реального времени.
4. Позволяет уменьшить расходы. Совершение сделки электронным путем на порядок снижает затраты на обслуживание сделки, а это, в свою очередь, влечет за собой снижение цен для покупателей.
5. Позволяет получать высококачественные услуги. Интернет-торговля позволяет продавцам повышать конкурентоспособность, становясь ближе к покупателю.
6. Уменьшает количество носителей информации, которые нужны для сохранения данных.
7. Сокращает процесс адаптации компании к изменениям рынка и время выхода товара на рынок .



8. Отсутствие таможенных налогов, связанных с продажами электронного контента.

9. Новые бизнес-модели: виртуальные предприятия, виртуальные агенты, технологии аутсорсинга и телеработы значительно повышают эффективность коммерческой деятельности. Кроме преобразования рынка существующих товаров и услуг, интернет-торговля открывает возможность появления совершенно новых продуктов и услуг. Например: страховые, брокерские услуги службы электронной поставки и поддержки.

10. Повышает к торговой марке уровень благосклонности потребителей. Качество обслуживания в интернете постоянно улучшается: покупатель может получить новую информацию о компании и товарах в любое удобное для себя время.

### **1.1.2 Новые возможности интернет-торговли**

Интернет-торговля позволяет поставщикам и потребителям в равной мере использовать новые возможности.

Глобальный выбор и присутствие: Поскольку интернет глобален, интернет-торговля позволяет даже самым мелким поставщикам достигать глобального присутствия и заниматься бизнесом в мировом масштабе. Также потребители получают глобальный выбор из всех потенциальных поставщиков, которые предлагают свои товары или услуги, независимо от их географического местоположения.

Повышение конкуренции и качество обслуживания: Интернет-торговля позволяет поставщикам конкурировать между собой, становясь ближе к потребителю. Соответственно, потребитель получает возросшее качество обслуживания.

Более подробная информация о спросе потребителей, персонализация товаров и услуг: Используя электронное взаимодействие, компании могут получать подробную информацию о запросах каждого потребителя и

автоматически предоставлять услуги и продукты , соответствующие их требованиям.

Сокращение доставки товара к потребителю, быстрая реакция на спрос:  
Интернет-торговля существенно позволяет улучшить товарные потоки. Товары доставляются напрямую от поставщика к потребителю в обход традиционных промежуточных пунктов в виде оптовых и розничных торговых компаний.

Уменьшение затрат, снижение цен: Оформление сделки электронным путем на порядок уменьшает затраты на обслуживание сделки, а это, в свою очередь, влечет за собой снижение цен для потребителей.

Наиболее комплексная, хотя и сложная в реализации, система Интернет-торговли – Интернет-магазин, в следующих главах речь пойдет об интернет-магазине.

## **1.2 Понятие интернет-магазина**

Интернет-магазин – это сайт (электронный ресурс), который предлагает посетителям для продажи товары и/или услуги. Такой сайт обычно представляет собой торговую-интернет-площадку, где выставлены изображения предлагаемых товаров или описания услуг, ресурс также снабжен инструментами для поиска, выбора и оформления заказа, имеется информация о самом магазине, контакты и информация для оплаты, доставки товаров или получения услуги.

Товары в интернет-магазине представлены в виде каталога, разделены на категории и имеют фотографии, описания, стоимость и другую информацию при необходимости. Интернет-магазин, как правило, обладает удобным поиском товаров по названию и другим ключевым словам. Посетитель сайта, просматривая информацию о товарах, имеет возможность выбрать заинтересовавший его товар и положить в "Корзину". Затем перейдя в

"Корзину" посетитель видит выбранные товары, где он может изменить количество товаров и удалить ненужные, далее посетитель может оформить заказ. Если на сайте есть личный кабинет, то авторизованный пользователь может видеть историю своих заказов, а так же их статус, например один из статусов заказа может быть: В ожидании, Доставляется или Доставлен.

Стандартный интернет-магазин состоит из следующих функциональных частей:

Каталог товаров — представляет собой сложную и многоуровневую структуру данных, которая должна простым и понятным способом выводить упорядоченные товары. Каталог представлен в виде дерева объектов, верхний уровень которого состоит из списка разделов. Разделы могут содержать подразделы или ссылки на конкретный товар и т.д. Такая структура данных просто необходима для удобного и быстрого нахождения товара.

Поисковая система — является обязательным элементом интернет-магазина. Несмотря на то что каталог обеспечивает упорядочивание и группировку данных, поисковая система дает пользователю возможность быстрого поиска информации, что особенно важно в том случае, когда каталог представляет собой достаточно разветвленную структуру данных с большим количеством разделов, подразделов и товаров, пользователь сложно представить в каком разделе может находиться интересующий его товар и есть ли он вообще в каталоге. Поисковая система во многих случаях позволяет существенно сократить количество переходов между страницами каталога для доступа к интересующей информации.

Пользовательская корзина — представляет собой некую ячейку, которая служит для хранения выбранного пользователем товара. В корзине можно изменить количество выбранного товара, удалить из корзины, а так же оформить заказ. Как и в любой корзине, интернет-магазина, пользователь может оформить заказ.

Форма оформления заказа – предназначена для ввода контактной информации заказчика и отправки ее и заказа на электронный ящик организации интернет-магазина.

Форма входа – служит для входа в личный кабинет, где пользователь может узнать о статусе своего заказа, изменить свои личные данные и т.д.

Форма регистрации – служит для регистрации пользователя в интернет-магазине, где он вводит свою контактную информацию.

### 1.3 Техническое задание интернет-магазина.

В следующих главах, будет приведено техническое задание на разработку интернет-магазина, а так же будут приведены методы разработки, структура сайта и описание технических частей.

#### 1.3.1 Используемые термины.

**Атрибут** – неотъемлемое свойство сущности (см. ниже).

**Контент** – публикуемая информация.

**Сущность** – единица структуры данных, объединяющая определенный класс объектов системы.

**Система управления сайтом CMS** – представляет собой специальный web-интерфейс, с помощью которого можно управлять структурой сайта, и его содержимым. Добавлять новости, новые страницы, редактировать текущие материалы, добавлять товары в каталог.

**CSS** - формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

**GET данные** – это тип данных, которые передаются по HTTP запросу (см. ниже), эти данные вносятся в URL (см. ниже) после знака « ? ». пример:

/path/resource?param1=value1&param2=value2, где (/path/resource) путь к ресурсу, а (key1=value1&key2=value2) GET данные, состоит из ключа «key» и значения «value», разделяются знаком «&», которые передаются запрашиваемому ресурсу.

**HTTP** - (HyperText Transfer Protocol — «протокол передачи гипертекста»), служит для передачи произвольных данных между клиентом и сервером.

**HTML (верстка)** - стандартный язык разметки страниц в Интернете.

**HTML (форма)** - Часть HTML-страницы, предназначенная для взаимодействия с посетителем сайта. Представляет собой набор элементов (текстовых полей, выпадающих списков), посредством которых пользователь может ввести какую-либо информацию и отправить ее для обработки на сервер.

**POST данные** — так же как GET, применяется для передачи пользовательских данных заданному ресурсу. Обычно POST данные передаются посредством HTML-форм, с помощью метода POST загружают файлы на сервер.

**URL** - единообразный локатор (определитель местонахождения) ресурса. URL служит стандартизированным способом записи адреса ресурса в сети Интернет.

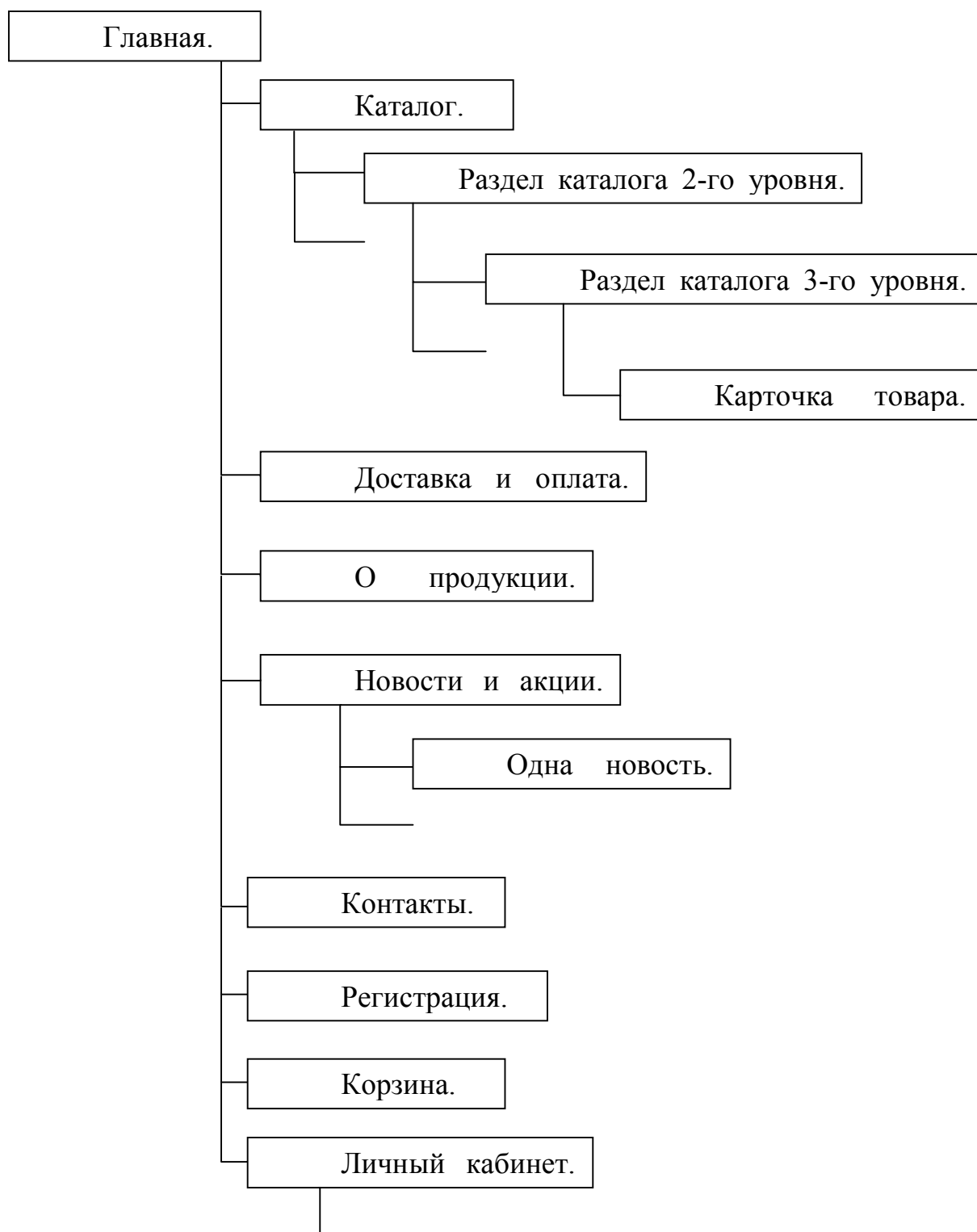
**WYSIWYG редактор** - Редактор языка HTML, имеющий возможности по работе в текстовом режиме и в режиме WYSIWYG (What You See Is What You Get). В редакторе WYSIWYG при редактировании в текстовом режиме, элементы HTML страницы подставляются автоматически.

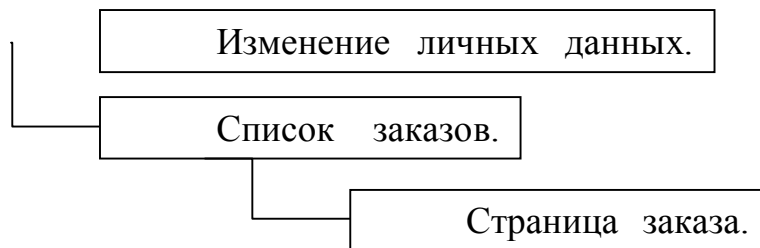
### 1.3.2 Выбор тематики интернет-магазина.

Тематика интернет-магазина выбрана мной, интернет-магазин называется – ВИТЯЗЬ, находится по веб адресу vityaz-tp.ru. Этот магазин предназначен для мужской аудитории, где предлагаются товары для рыбалки, охоты, туризма, а так же военный инвентарь. Дизайн мне предоставила web студия

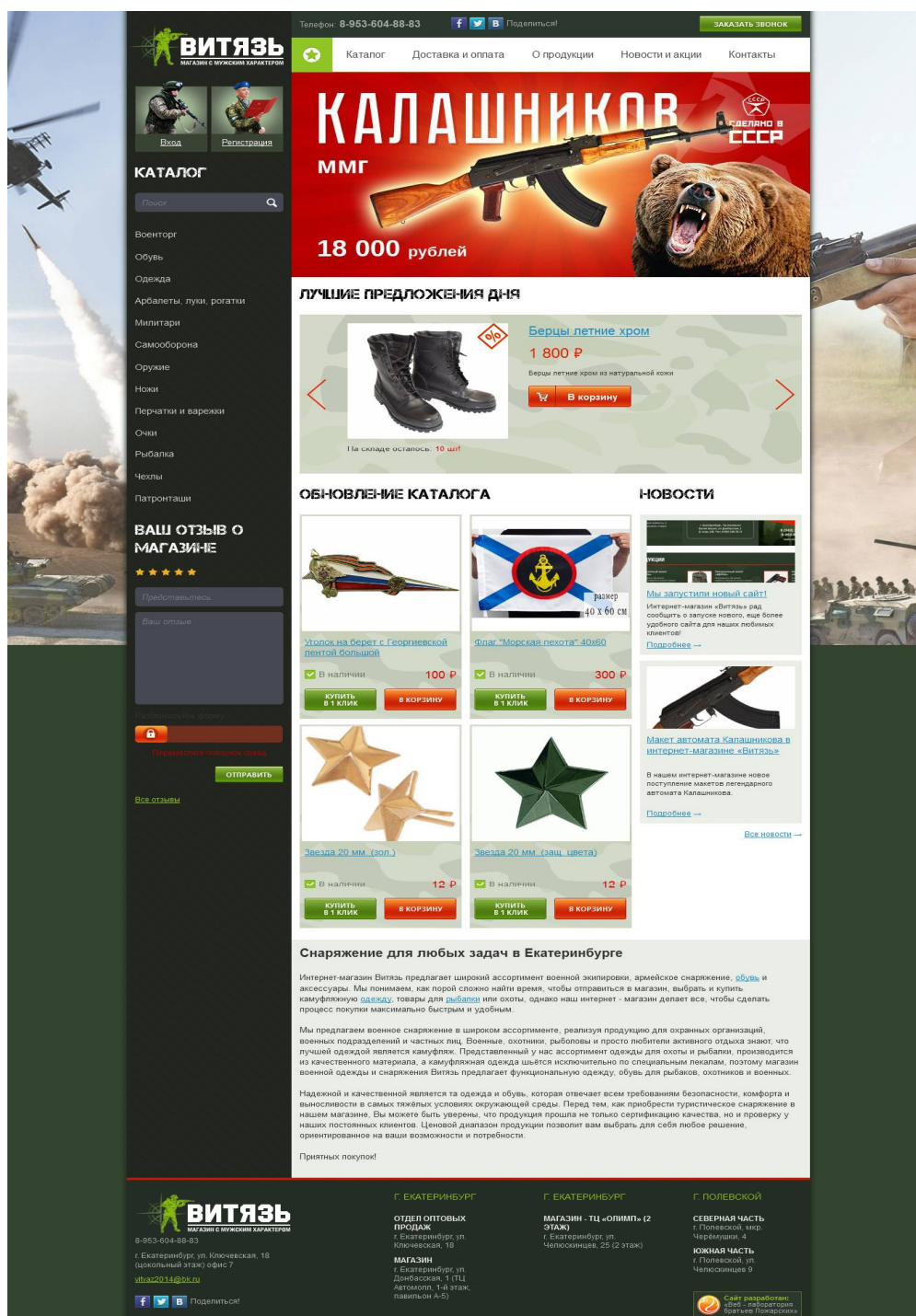
Братьев Пожарских, где собственно я работаю, а верстку и программирование сделана мной. Далее, будет более подробно описана структура интернет-магазина.

### 1.3.3 Структура интернет-магазина Витязь.





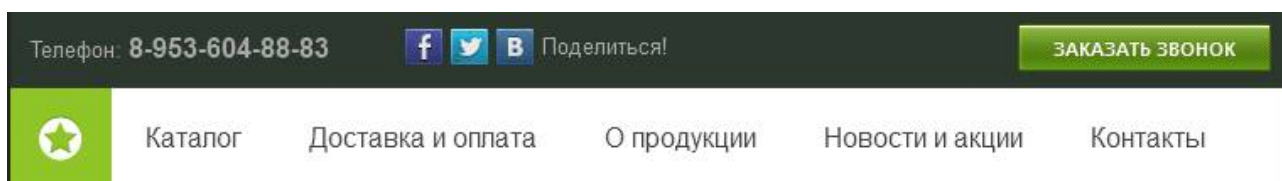
### 1.3.3.1 Главная



Главная страница интернет-магазина. Рис. 1

На (Рис. 1), изображение главной страницы интернет-магазина Витязь. Обычно у каждого веб-сайта есть контентные области: header (шапка сайта), footer (подвал сайта), content (основная информация страницы). Но эти термины не стоит ассоциировать с HTML тегами <header> и <footer>, тут речь идет о расположении информации на веб-странице. Header значит в верхней части веб-страницы, footer в нижней, а content где то между ними.

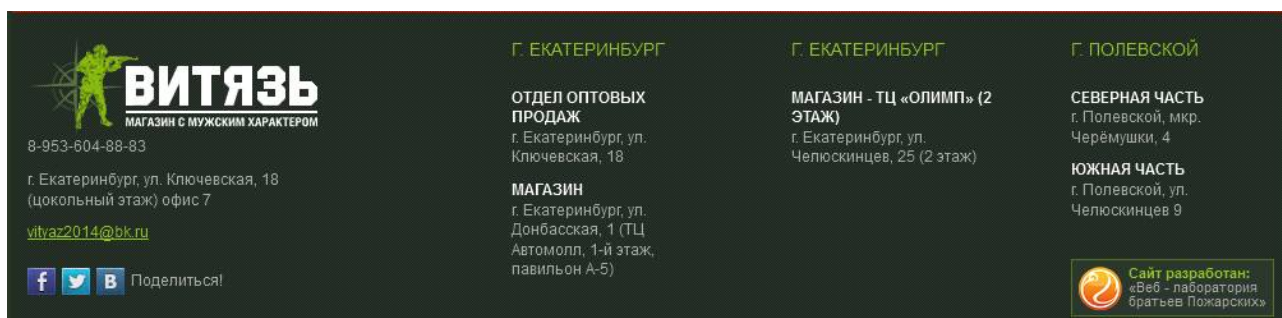
Header (Рис.1.2):



В header-е находится навигация, с помощью которой пользователь может переходить по разделам сайта. При переходе, на какой либо раздел, этот раздел становится активным. Так же в header-е указан телефон, ссылки на группы в социальных сетях, и кнопка «Заказать звонок», которая вызывает форму для ввода контактных данных (См. стр. 34). Контент в header-е остается неизменным. То есть при переходе по разделам сайта, только подсвечиваются активные разделы в навигации.

Footer

(Рис. 1.3):

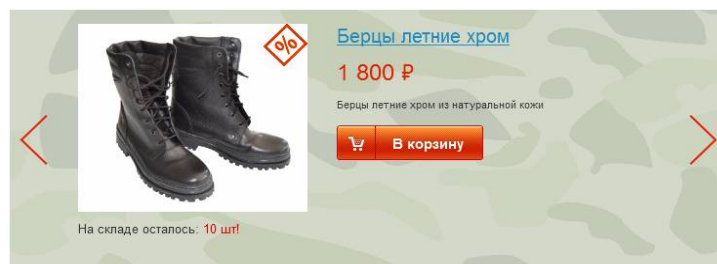




Footer (Рис. 1.3) содержит логотип компании, адреса магазинов,



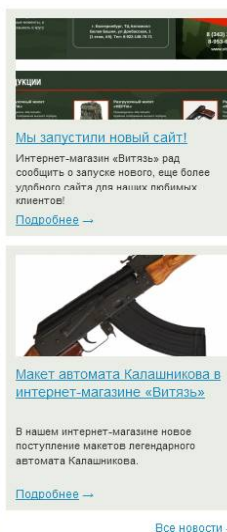
#### ЛУЧШИЕ ПРЕДЛОЖЕНИЯ ДНЯ



#### ОБНОВЛЕНИЕ КАТАЛОГА



#### НОВОСТИ



#### Снаряжение для любых задач в Екатеринбурге

Интернет-магазин Витязь предлагает широкий ассортимент военной экипировки, армейское снаряжение, обувь и аксессуары. Мы понимаем, как порой сложно найти время, чтобы отправиться в магазин, выбрать и купить камуфляжную одежду, товары для рыбалки или охоты, однако наш интернет - магазин делает все, чтобы сделать процесс покупки максимально быстрым и удобным.

Мы предлагаем военное снаряжение в широком ассортименте, реализуя продукцию для охранных организаций, военных подразделений и частных лиц. Военные, охотники, рыболовы и просто любители активного отдыха знают, что лучшей одеждой является камуфляж. Представленный у нас ассортимент одежды для охоты и рыбалки, производится из качественного материала, а камуфляжная одежда шьётся исключительно по специальным лекалам, поэтому магазин военной одежды и снаряжения Витязь предлагает функциональную одежду, обувь для рыбаков, охотников и военных.

Надежной и качественной является та одежда и обувь, которая отвечает всем требованиям безопасности, комфорта и выносливости в самых тяжёлых условиях окружающей среды. Перед тем, как приобрести туристическое снаряжение в нашем магазине. Вы можете быть уверены, что продукция прошла не только сертификацию качества, но и проверку у наших постоянных клиентов. Ценовой диапазон продукции позволит вам выбрать для себя любое решение, ориентированное на ваши возможности и потребности.

Приятных покупок!

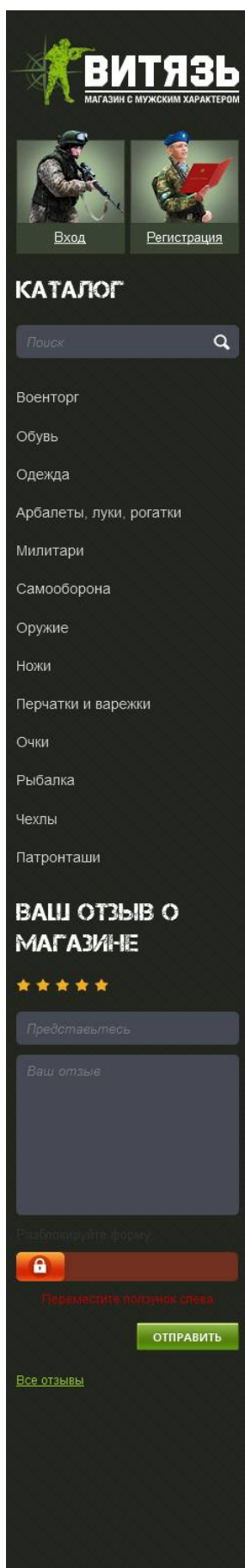
контактный телефон, e-mail, ссылки соц. сетей, а так же логотип разработчика. Содержимое в footer-е остается неизменным.

Content (Рис. 1.4):

Content, является основной контентной областью сайта. У каждой страницы, в этой контентной области, своя информация. В данном случае на Рис. 1.4 содержится информация, относящаяся к главной странице, это: «Горячие» и лучшие предложения, новые поступления, новости интернет-магазина, а также информация об интернет-магазине и товарах.

Из ранее перечисленных контентных областей, есть еще одна, назовем её

left-block.

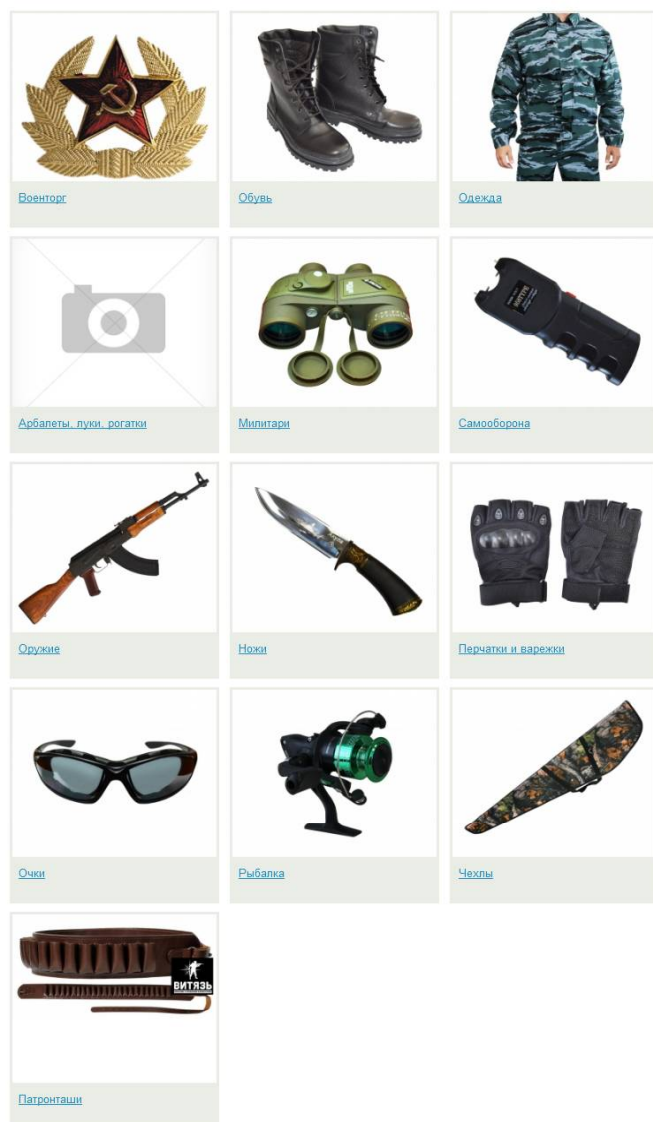


Left-block ([Рис. 1.5](#)):

Эта контентная область содержит логотип компании, ссылку «Вход», которая при нажатии вызывает форму для ввода логина и пароля, ссылку «Регистрация», которая ссылает на страницу регистрации, где для того чтобы зарегистрироваться на сайте, нужно заполнить необходимые поля. Ниже расположена, поисковая система, которая как описывалось ранее, является обязательным элементом интернет-магазина. Далее расположены разделы каталога первого уровня, как описывалось ранее, каталог это упорядоченная структура данных, где в первом уровне указывается тематика разделов, то есть по первому уровню можно определить, какие товары содержатся в разделах. Далее расположена форма отзывов. Для отправки отзывов пользователю сайта, независимо от того зарегистрирован он в системе или нет, может отправить свой отзыв. Для этого ему необходимо заполнить два поля, в первом поле пользователю нужно указать свое имя, во втором поле это соответственно сам отзыв, затем он может оценить свой отзыв по пятибалльной шкале, и в завершении перед отправкой нужно разблокировать форму, переместив ползунок с лева на право, иначе форма не отправится. Это нужно для того чтобы, защитить сайт от спама.

Остальные разделы сайта, будут показаны в контентной области «Content», так как это основная область для вывода информации.

## КАТАЛОГ



### Военная амуниция, одежда для охоты, рыбалки в одном месте

Наш интернет-магазин военной экипировки и армейского снаряжения постарался охватить интересы самых разных наших покупателей. Военная амуниция, в первую очередь, конечно, необходима сотрудникам силовых подразделений для успешного выполнения тактических задач. Тем не менее, камуфляж по-прежнему пользуется огромным спросом и у мирного населения. В первую очередь – это поклонники военизированных игр – в частности, пейнтбола, приверженцы стиля «милитари» в одежде и, конечно, рыбаки, охотники и туристы.

Одним из главных достоинств является то, что Вы можете, не вставая с дивана, посетить магазин военно-туристического снаряжения, выбрать, купить все необходимое для туризма, походов, охоты, рыбалки, пожарное снаряжение и многое другое.

Во-первых, ни одна боевая задача или поход не обойдется без надежной сумки или рюкзака. В нашем каталоге представлены камуфлированные сумки и рюкзаки, в том числе детские, изготовленные из прочных современных материалов. Инновационный камуфляж различных расцветок не позволит демаскироваться в любых условиях, что идеально подойдет для охотника, солдата или игрока в пейнтбол. Особое внимание следует уделить камуфлированным костюмам различных расцветок – НАТО, цифра, камыш и многие другие, позволяющие вам стать человеком-невидимкой в любых условиях.

Походный инструмент, представленный в нашем магазине военно-туристического снаряжения необходим каждому экстремалу, поскольку именно от него подчас зависит выживание в суровых условиях дикой природы. «Битля» – уникальный в своем роде магазин, камуфляж и снаряжение в котором рассчитаны на всех наших клиентов, будь то боевое, туристическое, горное, охотничье или рыболовное снаряжение. Экипировка на любой, даже самый высказательный вкус, по самым демократичным ценам и оформление в кратчайшие сроки – вот основные принципы работы нашего интернет-магазина военной экипировки и армейского снаряжения.

Если Вы решили купить тактическое или туристическое снаряжение оптом, не стоит тратить уйму времени на поиски оптимального поставщика. В нашем магазине туристического снаряжения продажа экипировки ведется как оптом, так и в розницу. Если опт окажется достаточно крупным, достаточно оформить предзаказ и нужное количество товара будет доставлено в кратчайшие сроки.

Качественное тактическое снаряжение – вопрос безопасности и комфорта военно-походной жизни для солдата.

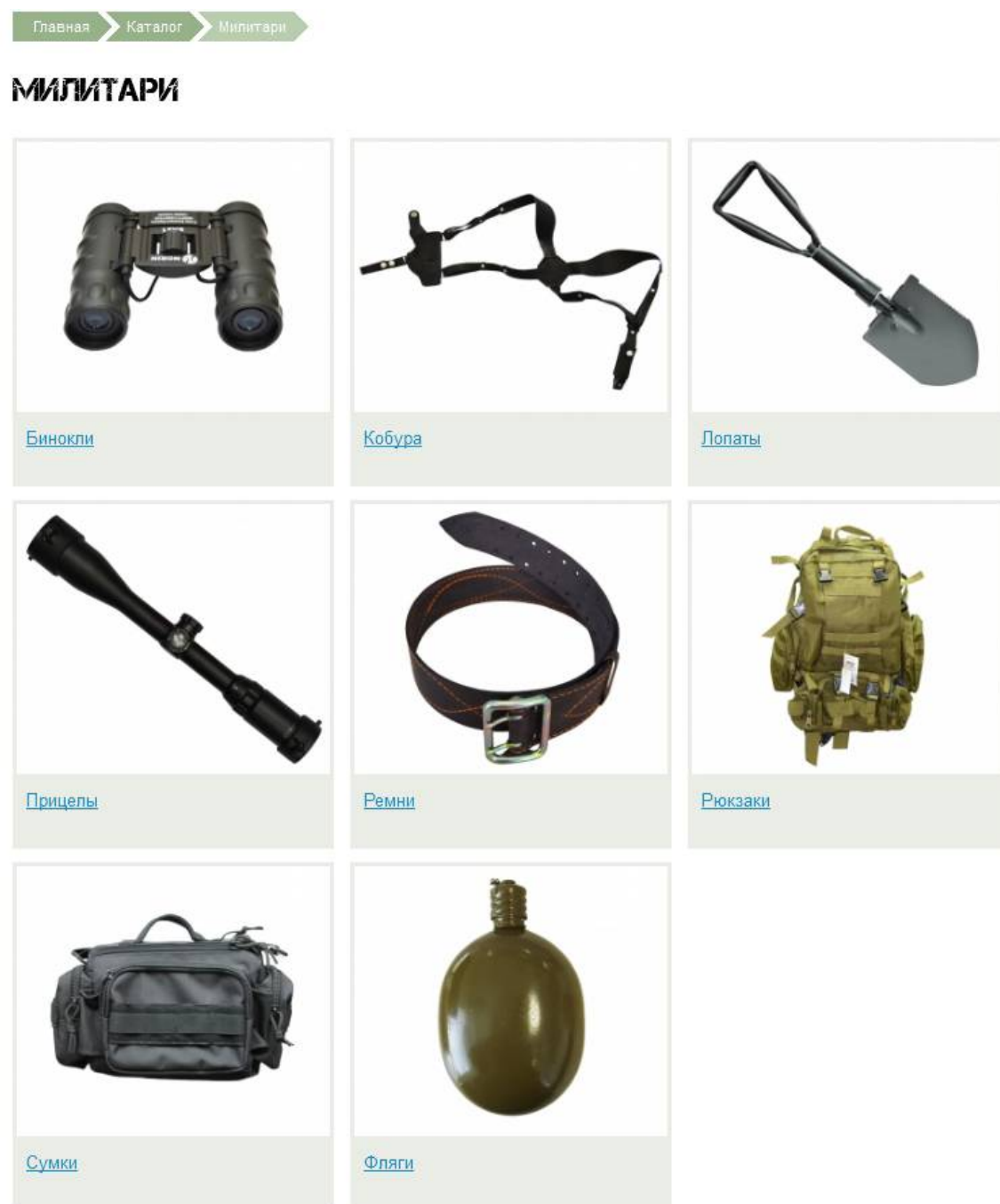
### 1.3.3.2 Каталог, разделы каталога первого уровня

Каталог, первый уровень ([Рис. 2](#)):

На [Рис. 2](#) представлены разделы каталога первого уровня, эти разделы также расположены в контентной области left-block, каждый из этих разделов содержат разделы второго уровня. Это древовидная структура данных, где корнем дерева служит каталог, который содержит разделы первого уровня, а разделы первого уровня, содержат разделы второго, о которых будет описано далее.

### 1.3.3.3 Каталог, разделы каталога второго уровня

Каталог, второй уровень ([Рис. 3](#)):





На [Рис. 3](#) расположены разделы каталога второго уровня с выбранной тематикой «Милитари». В каждом из этих разделов содержится список товаров.

#### 1.3.3.4 Каталог, разделы каталога третьего уровня


Каталог, третий уровень ([Рис.4](#)):

[Главная](#) > [Каталог](#) > [Милитари](#) > [Бинокли](#)

## БИНОКЛИ

Сортировать по:


▼



[Бинокль Bostron 10x50](#)

✓ В наличии 8 000 ₽


[КУПИТЬ В 1 КЛИК](#) [В КОРЗИНУ](#)



[Бинокль Veber Sport 8-17x25](#)

✓ В наличии 1 600 ₽

[КУПИТЬ В 1 КЛИК](#) [В КОРЗИНУ](#)



[Бинокль Norin 8x21](#)

✓ В наличии 1 000 ₽

[КУПИТЬ В 1 КЛИК](#) [В КОРЗИНУ](#)

В этом разделе находится список товаров. Над списком товаров расположен фильтр, где можно отсортировать список по дате или по цене. У каждого раздела товара (если он в наличии), расположены две кнопки это: «Купить в 1 клик» и «В корзину». При нажатии на «Купить в 1 клик», появляется форма (См. тамто), где пользователь вводит свою контактную информацию и подтверждает свой выбор. При нажатии «В корзину», товар попадает в корзину (См. тамто), в которую необходимо перейти, чтобы

оформить свой заказ. Если пользователь кликнет на изображение товара или на ссылку, то он попадет на страницу «Карточки товара».

### 1.3.3.5 Карточка товара

Карточка товара (Рис.5):

[Главная](#) > [Каталог](#) > [Милитари](#) > [Бинокли](#) > [Бинокль Bostron 10x50](#)



## Бинокль Bostron 10x50

Артикул: Бинокль Bostron 10x50

✓ В наличии

Цена: **8 000₽**


Бинокль Bostron 10x50

[КУПИТЬ В 1 КЛИК](#)[В КОРЗИНУ](#)



[Facebook](#)[Twitter](#)[ВКонтакте](#)


### ПОХОЖИЕ ТОВАРЫ



[Бинокль Norin 8x21](#)

✓ В наличии **1 000 ₽**

[КУПИТЬ В 1 КЛИК](#)[В КОРЗИНУ](#)



[Бинокль Veber Sport 8-17x25](#)

✓ В наличии **1 600 ₽**

[КУПИТЬ В 1 КЛИК](#)[В КОРЗИНУ](#)

На странице «Карточки товара», содержится более подробная информация о товаре, дополнительные изображения, ссылки социальных сетей, благодаря которым можно поделиться с друзьями. Под информацией о товаре, расположен список похожих товаров, в этом списке выводятся остальные товары из текущего раздела, в нашем случае это раздел «Бинокли».

### 1.3.3.6 Доставка и оплата

Доставка и оплата ([Рис.6](#)):



## ДОСТАВКА И ОПЛАТА

При заказе товара на сумму от 1000 рублей, доставка бесплатная. В случае заказа на сумму меньше 1000 рублей, стоимость доставки будет составлять 350 рублей.

- в близлежащие населённые пункты Екатеринбурга (В. Пышма, Березовский, Арамиль): - 450 руб. (доставка в эти города только при заказе от 1000 рублей).

Оплата производится, как наличным так и безналичным расчетом.



На этой странице подробная информация о доставке и оплате товара. Под текстом ссылка в каталог товаров.

### 1.3.3.7 О продукции

О продукции ([Рис.7](#)):

Главная > О продукции

## О ПРОДУКЦИИ



Рассматривая продукцию интернет-магазина «Витязь», нельзя не заметить явный мужской характер, который прослеживается в каждой единице товара. Их тематика и предназначение обширны. Ассортимент варьируется от аксессуаров для стрельбы до оружия различного вида. Он включает как снаряжение известнейших мировых производителей, так и своего производства.

Любители крупного улова найдут все, что им нужно для рыбалки в любой сезон. У нас есть садки, катушки, грузила, кормушки и прочие вспомогательные элементы. Обратите внимание на описание товара, оно поможет оценить преимущества конкретной модели и сделать правильный выбор. Поймать увесистую рыбку теперь не составит труда. Все для рыбалки и кемпинга смотрите в

соответствующей категории и выбирайте то, что наиболее Вам подходит и пришлось по душе.

Товары для охоты в любой сезон позволят с легкостью подстрелить зайца или другого лесного зверька. Можно даже попытать удачу на дикого кабана. Помимо ружей у нас представлена и специальная форма, она станет отличной маскировкой. Большой выбор пневматического оружия не оставит в покое настоящего мужчину. При этом мы не стоим на месте и регулярно обновляем магазин, поэтому постоянные покупатели нередко увидят интересные новинки.

Чего только стоит снаряжение для страйкбола, ведь в Екатеринбурге не так то и развита данная индустрия. Тактические очки, оружие и иные приспособления высокого качества разрешат насладиться игрой. С их помощью Вы окунетесь в невероятный мир борьбы и противостояния. Сможете командными силами победить противника и быть на высоте. Коллекционеров порадует большой выбор ножей. Самые неординарные рукоятки и изобретательные клинки станут украшением или даже главным акцентом коллекции.

Магазин с мужским характером также предоставляет возможность выбрать достойный подарок отцу, брату или любимому человеку. При возникновении каких-либо вопросов рекомендуем обратиться к менеджерам. Звоните по телефонам 8-343-32-773-98, 8-953-60-488-83 либо приходите в «Витязь». Все контакты прописаны на сайте.



На этой странице информация о продукции и об интернет-магазине «Витязь».





### 1.3.3.8 Новости и акции

Новости и акции ([Рис.8](#)):

## НОВОСТИ И АКЦИИ

2016

 <p><a href="#">Мы запустили новый сайт!</a> 02 Октября 2016 Интернет-магазин «Витязь» рад сообщить о запуске нового, еще более удобного сайта для наших любимых. <a href="#">Подробнее→</a></p>	 <p><a href="#">Макет автомата Калашникова в</a> 02 Октября 2016 В нашем интернет-магазине новое поступление макетов легендарного автомата Калашникова. <a href="#">Подробнее→</a></p>
--	--

На этой странице список новостей. При клике на ссылку или изображение переходим на страницу одной новости.

### 1.3.3.9 Одна новость

Одна новость ([Рис.9](#)):

## Мы запустили новый сайт!



Вас ожидает большой каталог товаров, как новые поступления, так и уже полюбившиеся вам и проверенные временем вещи. Следите за обновлением каталога и вы обязательно найдете что-нибудь интересное для себя!



На странице, содержится подробная информация одной из выбранных новостей.

### 1.3.3.10 Контакты

Контакты ([Рис.10](#)):

На странице «Контакты», список адресов. У каждого адреса есть своя

карта (Яндекс карта),

заголовок, адрес, номер

телефона. Адреса

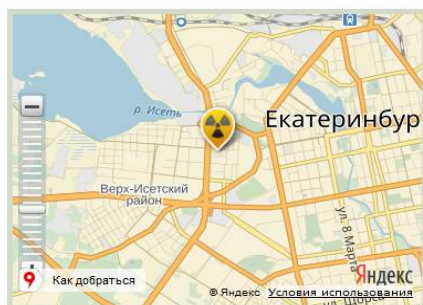
сгруппированы по

названию города.

Главная Контакты

## КОНТАКТЫ

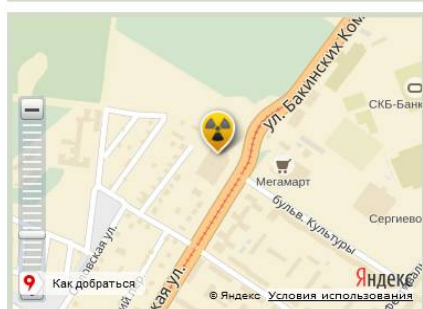
г. Екатеринбург



### ОТДЕЛ ОПТОВЫХ ПРОДАЖ

Адрес: г. Екатеринбург, ул. Ключевская, 18

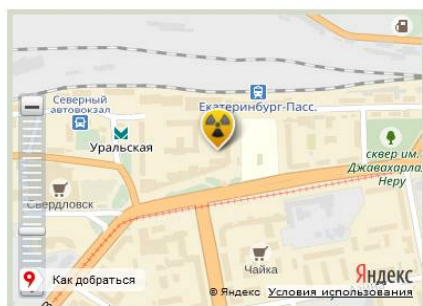
Телефон: 8 (343) 327-73-98



### МАГАЗИН

Адрес: г. Екатеринбург, ул. Донбасская, 1 (ТЦ Автомолл, 1-й этаж, павильон А-5)

Телефон: 8 953 604 88 83

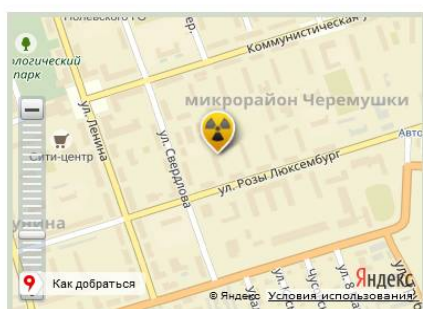


### МАГАЗИН - ТЦ «ОЛИМП» (2 ЭТАЖ)

Адрес: г. Екатеринбург, ул. Челюскинцев, 25 (2 этаж)

Телефон: 8 950 563 81 00

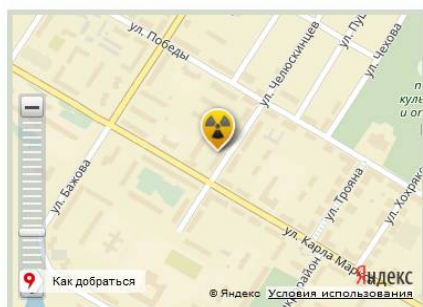
г. Полевской



### СЕВЕРНАЯ ЧАСТЬ

Адрес: г. Полевской, мкр. Черёмушки, 4

Телефон: 8 953 601 32 23



### ЮЖНАЯ ЧАСТЬ

Адрес: г. Полевской, ул. Челюскинцев 9

Телефон: 8 (343-50) 2-06-15

### 1.3.3.11 Регистрация

Регистрация ([Рис.11](#)):



The registration form is displayed on a page with a green header bar containing two buttons: 'Главная' (Home) and 'Регистрация' (Registration). The title 'Регистрация' is prominently displayed above the form fields. The form consists of several input fields, each preceded by a label and a red asterisk indicating a required field. The fields are: 'Фамилия' (Surname), 'Имя / Отчество' (Name / Patronymic), 'Адрес проживания / Платежный адрес' (Residence address / Billing address), 'Телефон' (Phone), 'E-Mail', 'Пароль (от 6 символов и без спец. знаков)' (Password (6+ characters, no special characters)), and 'Подтверждение пароля' (Confirm password). Below these fields is a section for 'Разблокируйте форму' (Unlock the form), which includes a red slider control with a lock icon and the instruction 'Переместите ползунок слева направо >' (Move the slider from left to right >). At the bottom of the form, there is a button with a red 'X' icon and the text 'Очистить все поля' (Clear all fields), and a large orange button labeled 'ОТПРАВИТЬ' (SEND).

Главная Регистрация

## Регистрация

Фамилия: \*

Имя / Отчество: \*

Адрес проживания / Платежный адрес: \*

Телефон: \*

E-Mail: \*

Пароль (от 6 символов и без спец. знаков): \*

Подтверждение пароля: \*

Разблокируйте форму:

Переместите ползунок слева направо >

Очистить все поля

ОТПРАВИТЬ

Страница регистрации содержит форму. Для того чтобы зарегистрироваться на сайте, нужно заполнить все необходимые поля, разблокировать форму и нажать на кнопку отправить. Поля ввода обладают интерактивностью, то есть при вводе данных, поля сразу проверяются на валидность данных. Например: поле E-mail при вводе сразу проверяется

(действительно ли введен E-mail, а никакой ни будь набор символов), если введенные данные удовлетворяют условию, то перед отправкой формы, E-mail снова проходит проверку (не регистрировался ли ранее пользователь с таким же E-mail), если да, то появляется сообщение об ошибке (E-mail уже зарегистрирован). Такая строгость объясняется тем что, E-mail будет служить не только для уведомлений на электронную почту, но и так же E-mail будет служить логином при входе в личный кабинет. Все поля ввода, помеченные красной «звездочкой», обязательно должны быть заполнены. При проверке поля ввода на валидность, рамка поля меняет свой цвет на красный или зеленый, извещая о том что, верно ли введены данные? если верно то цвет рамки становится зеленым, если нет, то соответственно красным. Так же поля формы можно моментально очистить, для этого нужно нажать на «ссылку» в левом нижнем углу (Очистить все поля).





#### 1.3.3.12 **Корзина**

Корзина это необходимая часть любого интернет-магазина, как описывалось ранее ([См. стр. 11](#)). У каждого товара (если он в наличии), есть кнопка (Добавить в корзину), при нажатии на которую, товар попадает в корзину. Страница корзины доступна только тогда, когда в ней есть товары. Если пользователь авторизован, то товары в корзине синхронизируются с его учетной записью, то есть пользователь, положив товары в корзину, может покинуть сайт на долгое количество времени, и затем, вернувшись в интернет-магазин и войдя в личный кабинет, его ранее заполненная корзина восстановится, и тогда он сможет продолжить свои покупки. А для не авторизованных пользователей, товары в корзине будут лежать несколько дней.

## Корзина ([Рис.12](#)):

[Главная](#) > [Оформление заказа](#)

### Оформление заказа

	Звезда 20 мм. (зол.)	<div>- 2 +</div>	12 руб 24 руб	✕
	Берцы зимние Рейнджер	<div>- 1 +</div>	2 800 руб	✕
	Ботинки рабочие ВА 412 УТ	<div>- 1 +</div>	1 500 руб	✕
	Лук	<div>- 1 +</div>	5 700 руб	✕

[Очистить корзину](#)

ДОБАВИТЬ ЕЩЕ ПОЗИЦИЮ

Итого: **10 024Р**

Фамилия:

тест

Имя / Отчество:

тест

Ваш адрес:

тест

Телефон: \*

+7 (456) 457-54-74

Вариант доставки:

Доставка курьером

Комментарии к заказу:

ОФОРМИТЬ ЗАЯВКУ



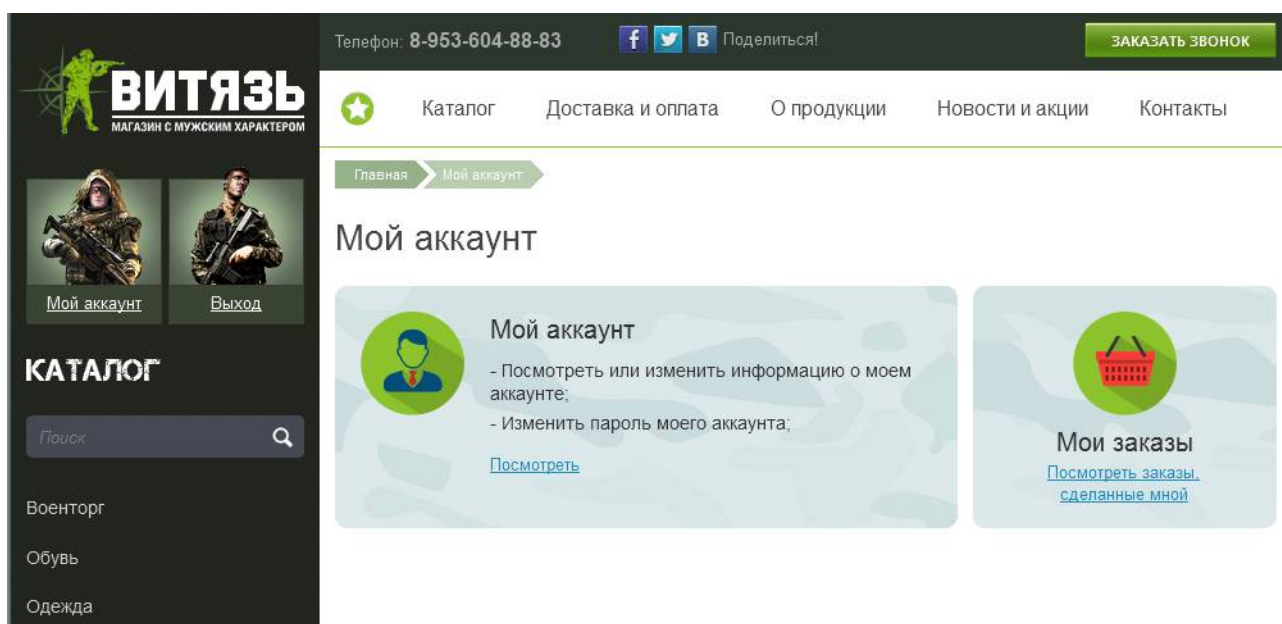
В корзине уже выбраны для демонстрации несколько товаров. Список выбранных товаров, представлен в табличном виде, где у каждого товара своя строка. В первой ячейке изображение и заголовок товара, во второй ячейке количество и кнопки для изменения количества товара, в третьей ячейке цена и сумма товара, сумма динамически обновляется при изменении количества товара, и наконец, в последней ячейке кнопка в виде крестика, с помощью



которой, можно удалить товар из списка. Сразу под таблицей расположены кнопки для очистки корзины, переход в каталог товаров и итоговая сумма. Еще ниже размещена форма, в которой нужно заполнить необходимые поля перед оформлением заказа. Если пользователь авторизован, то вся необходимая контактная информация заполняется автоматически, при желании пользователь может изменить контактные данные и ввести дополнительную информацию. Заявка оформляется при нажатии на кнопку «Оформить заявку».

### 1.3.3.13 Личный кабинет

Личный кабинет ([Рис.13](#)):



Личный кабинет доступен после авторизации пользователя. В личном кабинете доступны две ссылки, это: «Мой аккаунт», где можно изменить личные данные и «Мои заказы», где история заказов. В Left-block, в верхней части доступны ссылки в личный кабинет и выход из личного кабинета.

### 1.3.3.14 Изменение личных данных

Изменение личных данных ([Рис.14](#)):

Главная > Мой аккаунт > Изменить личные данные

## Изменить личные данные

Фамилия: \*

Имя / Отчество: \*

Адрес проживания / Платежный адрес: \*

Текущий пароль: \*

Новый пароль (от 6 символов и без спец. знаков): \*

Подтверждение нового пароля: \*

СОХРАНИТЬ

На этой странице пользователь может изменить свои данные, эти же данные подставляются в форму оформления заказа.



### 1.3.3.15 Список заказов

Список заказов ([Рис.15](#)):

Главная > Мой аккаунт > Мои заказы

## Мои заказы

Номер заказа: **56** Статус заказа: Не подтвержден

Дата заказа: 02.10.2016	Товаров: 2	<a href="#">СМОТРЕТЬ</a>
Получатель: тест	Сумма счета: 600 руб.	

Номер заказа: **55** Статус заказа: Не подтвержден

Дата заказа: 02.10.2016	Товаров: 5	<a href="#">СМОТРЕТЬ</a>
Получатель: тест	Сумма счета: 10 024 руб.	

[← Назад](#)

На этой странице список ранее оформленных заказов.

### 1.3.3.16 Страница заказа

Список заказов ([Рис.16](#)):

Главная > Мой аккаунт > Мои заказы > Заказ №55

## Заказ №55

Дата заказа: **02.10.2016** (Не подтвержден) Итого заказ: **10 324 руб.**

Товары		
1.	<a href="#">Звезда 20 мм. (зол.)</a>	12 руб. 2 шт. 24 руб.
2.	<a href="#">Берцы зимние Рейнджер</a>	2 800 руб.
3.	<a href="#">Ботинки рабочие ВА 412 УТ</a>	1 500 руб.
4.	<a href="#">Лук</a>	5 700 руб.
		Стоимость: <b>10 024 руб.</b>
Доставка курьером (доставляется в пределах города Екатеринбурга, обычно в день заказа (или на следующий день)):		<b>300 руб.</b>
		Итого: <b>10 324 руб.</b>

Адрес доставки:  
тест  
тест

Вариант доставки:  
Доставка курьером: Доставка курьером (доставляется в пределах города Екатеринбурга, обычно в день заказа (или на следующий день))

Метод оплаты:  
:

[← Назад](#)

Страница одного заказа, это уже оформленный заказ, с подробной информацией.

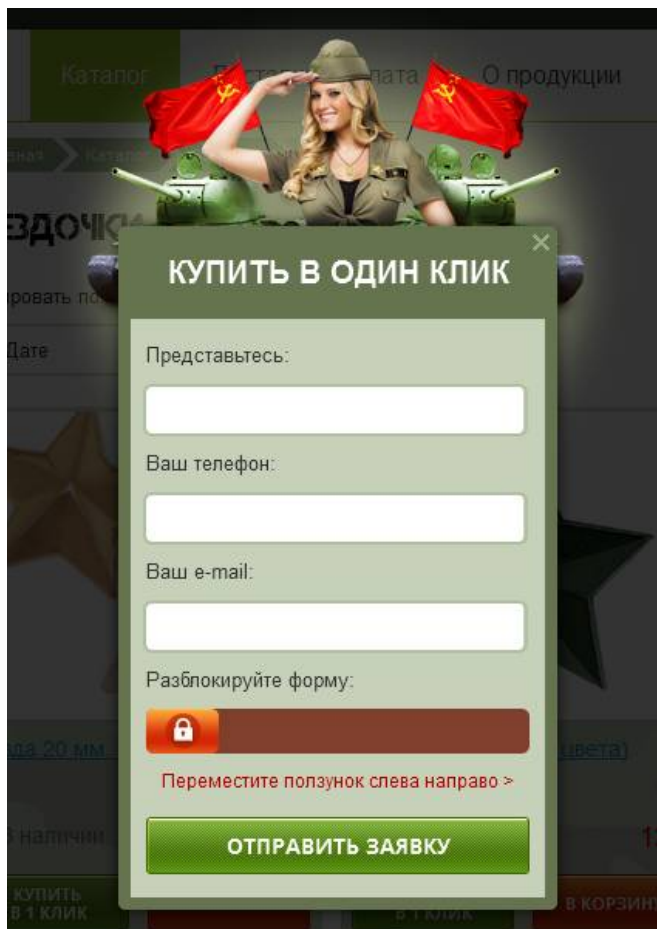
### 1.3.3.17 Всплывающие формы

Форма заказать звонок ([Рис.17](#)):

С помощью этой формы, пользователь сайта может заказать звонок. Пользователь вводит имя и телефон, после отправки формы, в ближайшее время с ним связывается менеджер сайта.

Форма входа в личный кабинет ([Рис.17.1](#)):

Эта форма для входа в личный кабинет. Пользователь вводит свой логин и пароль и если введенные данные верны, то пользователь входит в свой личный кабинет.



Форма купить в один клик (Рис.17.2):

Благодаря этой форме, можно оформить покупку намного быстрее, чем традиционным способом, где перед оформлением заказа, требуется положить товар в корзину, затем перейдя в корзину, только тогда можно оформить покупку. А при помощи формы «Купить в один клик», пользователь оформляет заказ следующим образом: Когда пользователь находит желаемый товар, он кликает на кнопку «купить в 1 клик» и сразу же появляется

форма оформления заказа, на которой пользователь заполняет все необходимые данные и нажимает на кнопку «Оформить заявку».

### 1.3.4 Метод разработки.

Прежде чем будет описан метод разработки интернет-магазина «Витязь», будут приведены все основные методы создания сайтов.

Существует несколько методов разработки сайтов. Начиная с готовых решений и заканчивая программированием с нуля. В каждом из методов разработки сайтов есть свои преимущества и недостатки, которые далее будут рассмотрены.

1. Конструктор сайтов : Эта система предоставляет набора инструментов, где без каких-либо особых знаний, можно создать и администрировать

сайт. С помощью конструктора можно выбрать тип сайта (сайт визитка, интернет-магазин и т. д.), шаблон (готовый дизайн сайта) и модули, которые помогут расширить функциональность сайта. Одни из самых популярных конструкторов сайтов это: Wix, ukit, uCoz и многие другие.

Основные преимущества конструкторов сайтов это:

- Простота в использовании. Не требуется определенных знаний в веб разработках, все делается пошагово и легко.
- Дизайн отделен от текстовой части. В несколько кликов можно изменить дизайн или изменить текст, не опасаясь за «поломку» сайта.
- Готовые шаблоны дизайна. Шаблон можно просто и легко заменить в пару кликов.
- Экономия времени и денег. Возможность создать сайт без лишних затрат и за небольшое количество времени.

Из недостатков конструкторов сайтов это:

- Реклама. Если была применена бесплатная версия, то на сайте будет навязана реклама, которая снизит прибыль сайта, ведь на сайте может быть реклама конкурентов.
- Ограниченно дисковое пространство. Многие бесплатные конструкторы выделяют слишком мало места на жестком диске. Которое, со временем заканчивается, и приходится ограничивать себя в пополнении каких либо файлов.
- Ограниченный функционал. Не каждый конструктор может похвастаться широким набором инструментов. Обычно набор функциональных возможностей стандартный и ограниченный.

- Отсутствие индивидуальных решений. Если потребуется внести, какие ни будь индивидуальные решения на сайт, то этого сделать не получится.

2. Готовые CMS системы: CMS расшифровывается как - Content management system, в переводе – Система управления содержимым. Готовая CMS означает что - она уже готова для запуска в интернете, к ней прилагаются разнообразные модули, шаблоны дизайна. Остается только установить её на веб сервер, выбрать подходящую тему оформления, заполнить данными и сайт уже готов. Для дизайна сайта используются готовые шаблоны, которые можно редактировать (изменяя HTML, CSS) под свои нужды. У каждой CMS системы есть панель Администрирования, к которой нужен доступ. С помощью этой панели, можно изменить структуру сайта, задействовать определенные модули, изменить данные и многое другое. Особой популярностью пользуются Joomla, Wordpress, Drupal, Битрикс и так далее...

Преимущества готовых CMS систем это:

- Готовый продукт. Безусловно, самое большое преимущество готовых CMS систем: не нужно ничего программировать с нуля – всё уже готово для использования. Например, если понадобится быстро запустить свой сайт (интернет-магазин, сайт визитку, либо что другое), то под любую задачу есть конкретная готовая CMS и это очень удобно.
- Простота установки. Готовую CMS систему практически легко и быстро установить на веб сервер. Вся установка происходит заливкой (размещением файлов) готовой CMS на веб-сервер, после этого необходимо запустить файл установки, где далее обычно необходимо ввести необходимые данные, а именно: доступы для подключения к базе данных, имя сервера, логин, пароль и т.д. Далее

CMS сама создаст необходимые таблицы в базе данных, заполнит их необходимыми данными, после этого уже можно приступить к наполнению и оформлению нового сайта.

- **Функциональность.** Готовые CMS системы обладают огромным функционалом и достаточно продвинутой административной панелью. Разнообразие функционала, объясняется тем - чтобы можно было решить наибольшее количество задач.
- **Разнообразие дополнений.** Несмотря на предустановленные темы оформления и плагины, в интернете существует большое разнообразие готовых тем, плагинов и других расширений.
- **Популярность.** Готовые CMS системы пользуются огромной популярностью. В интернете можно найти решение о возникшей проблеме. Так же с популярными CMS системами работает немалое количество программистов, которые разрабатывают новые темы оформления и другие расширения.

Недостатки готовых CMS систем:

- **Избыточная функциональность.** В готовых CMS реализовано очень много функционала, это ведет к более долгому формированию веб-страницы и неоправданно большой базе данных. Для реализации необходимых задач, используется примерно 30% функционала готовой CMS – остальные 70% не используются вовсе.
- **Низкая безопасность.** Готовые CMS системы очень популярны, их исходный код доступен каждому. В интернете есть немало статей, как взломать Joomla или Wordpress, и это создает существенные проблемы с безопасностью.
- **Проблемы с расширяемостью.** Несмотря на изобилие предлагаемых всевозможных расширений для готовых CMS систем. Всегда возникаю нетривиальные задачи, например: сделать три цены у товара, или изменение интерфейса под определенные

поведенческие факторы посетителей сайта. Такие случаи приводят к утомительному изучению ядра сайта для внесения изменений.

- Сложность. Готовые CMS содержат очень много нужных и ненужных настроек, которые усложняют использование готовой CMS системы.
- Поисковики не «любят» готовые CMS системы. Из-за медлительной загрузки сайта, не уникальности в структуре сайта (обычно используют шаблоны, которые уже где то используются) и прочих факторов, могут возникнуть проблемы продвижения (для поднятия в топы в поисковиках) в Интернете.

3. Самописная CMS система: Это система, разработанная одним или несколькими разработчиками, веб студией. Такие системы разрабатывают по разным причинам: привязать клиента к конкретной веб студии, попытка сделать продукт который должен по некоторым показателям превосходить готовые CMS, создать CMS систему для самоутверждения. Есть два типа самописных CMS: первый тип - это разработанные полностью с нуля (не прибегая, к каким ни будь инструментам), второй тип - это разработанные на основе фреймворков (см. тамто) и других веб-инструментах.

Преимущества самописных CMS систем:

- Скорость работы. Самописные CMS системы отличаются высокой скоростью работы. Из-за отсутствия не нужного функционала и создания системы под конкретные условия.
- Решение нестандартных задач. На CMS собственной разработки, намного быстрее и проще реализовать оригинальную идею.
- Безопасность. Самописные CMS системы наиболее безопасны, потому что исходный код знает только разработчики.
- Недостатки самописных CMS:

- Привязка к программисту. Сайт создается конкретным программистом, другому разработчику будет сложнее разобраться в коде. Но это зависит от уровня разработчика и от типа самописной CMS. Если система написана на основе фреймворка и в коде есть комментарии, то у среднего специалиста не должно возникнуть проблем.
- Длительная разработка. Из-за разрабатываемых с нуля многих функций, программисту необходимо больше времени.

Это были все основные методы разработки сайтов. Каждому, кому нужен сайт, найдет для себя подходящий метод, все зависит от требований и бюджета.

Интернет-магазин «Витязь» разработан на самописной CMS системе, то есть по третьему методу. Эта CMS, создана на основе фреймворка и других дополнительных инструментах, о которых пойдет речь в следующих главах. Следующие главы посвящены самописной CMS системе, на базе которой сделано немало сайтов, в том числе и интернет-магазин «Витязь».

## 1.4 Описание CMS системы

CMS системы была разработана мной в 2013 году, по настоящее время, на ней уже было сделано около 30 сайтов – это интернет-магазины, обычные информационные сайты, личные кабинеты, лендинги (одностраничники) и другие. За все время существования, CMS система потерпела немало изменений, и модернизаций. Эта система не имеет, каких либо готовых шаблонов дизайна, или других готовых решений. Она предназначена для разработки сайтов с нуля.

В следующих главах речь пойдет об используемых технологиях и инструментах, которые были применены для создания CMS системы.

### 1.4.1 Веб-технологии CMS



Прежде чем будут описаны технологии и инструменты, которые были использованы для создания CMS системы. Будут приведены несколько терминов, которые помогут лучше понять взаимодействие веб технологий.

Front-end: Это клиентская часть, то есть все что находится на стороне клиента. К front-end относятся следующие веб технологии как - CSS, HTML, JavaScript – с помощью которых создается пользовательский интерфейс.

Инженеры, занимающиеся этой частью сайта, называются front-end разработчиками. Этих специалистов также называют веб-дизайнерами, верстальщиками.

Back-end: Это серверная часть, на которой находится ядро сайта, это ядро обеспечивает основной функционал и поддержку front-end. В back-end, используется один из серверных языков программирования такие как – PHP, ASP.NET, Python, Ruby, Perl и другие. Серверные языки программирования рассчитаны на работу с базами данных, такие как - MySQL, MongoDB и т.д. Инженеры, ответственные за разработку серверной части веб-сайта, называются back-end разработчиками.

Далее будут приведены веб технологии, на основе которых работает CMS система.

Back-end:

- PHP — это серверный язык программирования, специально разработанный для написания веб-приложений. Изначально он был разработан для отслеживания веб-посетителей. Со временем из набора инструментов, PHP превратился в полноценный ООП (Объектно ориентированный) язык программирования, его код может так же внедряться в HTML. При посещении пользователем

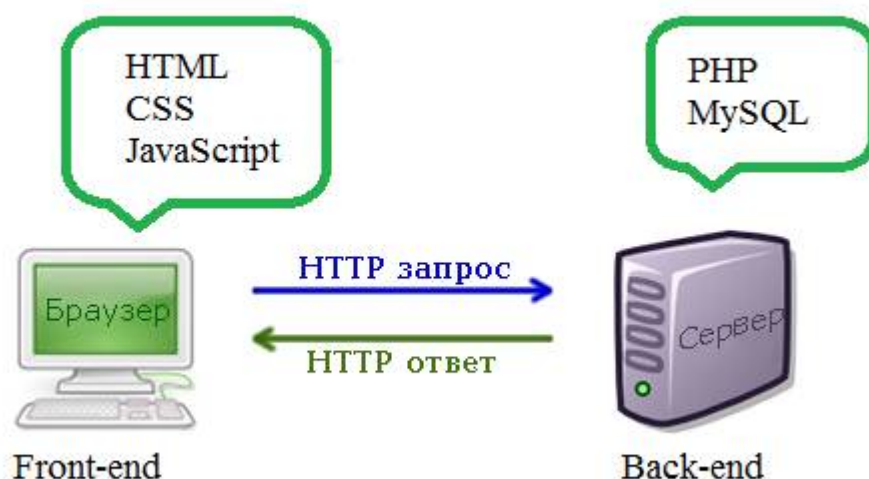
веб-сервера, PHP формирует HTML страницу и отправляет обратно в браузер пользователя.

- MySQL — это система управления базами данных (СУБД). База данных представляет собой структурированный набор данных. Она может содержать различную информацию — от простого списка покупок до огромного объема данных, используемого в корпоративной сети.

Front-end:

- HTML – Это язык разметки гипертекстовых документов. HTML описывает структуру веб-страницы с помощью тегов (HTML элементов), эти элементы определяют расположение текста, рисунка, таблиц и многое другое.
- CSS – это правила описывающие стиль (вид) HTML документа. С помощью CSS, веб-страница приобретает зрительно понятный интерфейс. То есть, CSS описывает правила для HTML тегов, текста, изображений, изменяя тем самым – расположение элементов относительно друг друга, размер, цвет, шрифт и другие визуальные изменения.
- JavaScript – Это язык программирования выполняемый на стороне клиента. Он разработан для того что бы выполнять какие либо программные действия в браузере, придавая тем самым интерактивность на веб-странице. JavaScript позволяет манипулировать HTML элементами, CSS стилями веб-страницы и много другое.

Для обмена данными между Front-end и Back-end, используется HTTP протокол. Это протокол для обмена информации в сети интернет.



Пример взаимодействия Front-end и Back-end. Рис. 18

На рис. 18, пример взаимодействия Front-end и Back-end. Когда пользователь посещает веб-ресурс, ему в ответ приходит веб-страница (Front-end), которая была сформирована сервером (Back-end).

### 1.4.2 Фреймворк CMS системы

Фреймворк – это программный продукт или инструмент или каркас, который упрощает разработку и поддержку технически сложных или нагруженных проектов. Фреймворк, предоставляет базовые программные модули, а все специфичные для проекта компоненты, реализуются разработчиком на их основе. Одни из самых популярных PHP-фреймворков для веб разработок это: Yii, Zend, Symfony, Kohana и другие. На основе фреймворка Kohana, была разработана CMS система.

Kohana – это ООП PHP фреймворк, использующий концепцию: HMVC (Hierarchical Model View Controller - Иерархические Модель-Вид-Контроллер см. далее). Его основные цели - быть безопасным, легким и простым в использовании. Его особенности это – Высокая скорость работы, безопасность, большое количество встроенных инструментов, простота понимания, использование концепции HMVC, полная совместимость с UTF-8 и другое.

MVC (Model View Controller – Модель Вид Контроллер) - Это один из самых известных, и часто применяемых шаблонов проектирования. Этот шаблон не зависит от языка программирования, а просто описывает технологию, модель или последовательность действий (алгоритм) для решения поставленной задачи.

Model (Модель): Используется для доступа и манипулированию данных. Обычно модель используется для доступа к хранилищу данных (например, к базе данных). Модель предоставляет интерфейс для поиска, создания, модификации и удаления данных из хранилища.

View (Вид): Это часть системы, которая содержит отображаемую информацию приложения. Чаще всего это HTML, CSS и JavaScript. Задача Вида - это отделение отображения информации от логики приложения для возможности повторного использования и чистоты кода.

Controller (Контроллер): Является прослойкой между Моделью и Видом в приложении. Когда контроллеру нужно изменить или получить данные, он обращается к Модели. Контроллер передает информацию, полученную из Модели, в Вид, где конечный результат показывается пользователю. Контроллеры, по сути, управляют распределением информации в приложении.

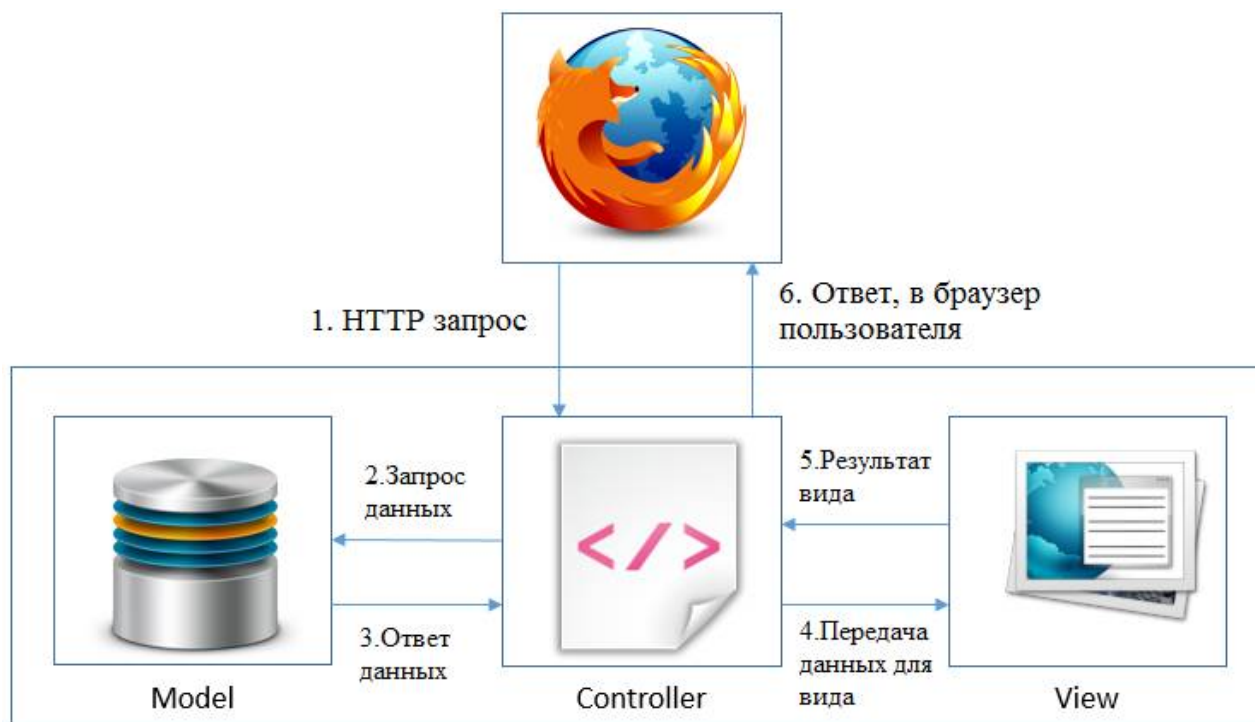


Схема взаимодействия MVC Рис. 19

На рис. 19, изображена типичная схема взаимодействия MVC структуры. Где http запрос поступает с браузера на сервер к контроллеру, контроллер для формирования результата обращается к модели за данными, затем контроллер передает полученные данные с модели в вид, далее вид формирует результат в виде веб-страницы и этот результат возвращается контроллером в браузер.

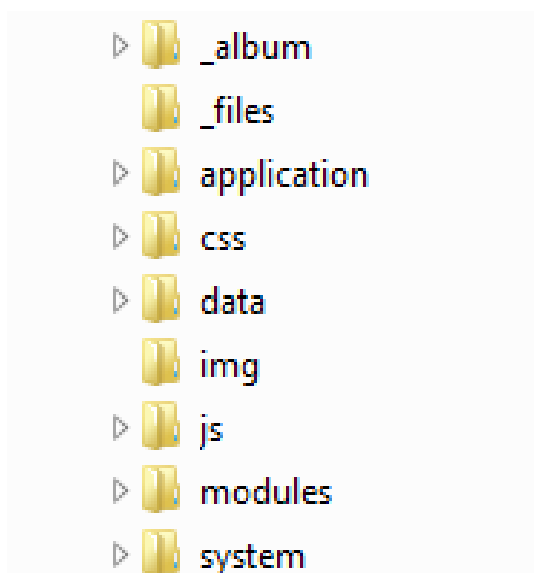
### 1.4.3 Файловая структура CMS

Далее рассмотрим файловую структуру CMS системы. Ранее было приведено о том, что CMS была, разработана на фреймворке Kohana. По сути, будет рассмотрена файловая структура фреймворка. Так как CMS система разработана в виде модуля, и она не вносит изменений в файловой структуре фреймворка.

## Файловая структура. Рис. 20

На рис. 20, приведена файловая структура фреймворка Kohana. Основные директории, которые входят в фреймворк это – application, modules и system. Остальные директории служат файловыми ресурсами. Сначала будут рассмотрены файловые ресурсы, а затем основные директории фреймворка.

- **\_album:** В этой директории содержатся изображения, сохраняемые CMS



системой. Через панель администрирования, пользователь загружает изображения, которые выводятся на веб-сайте. CMS система сохраняет каждое изображение в двух экземплярах. Первое изображение исходное, а второе в жатом виде (preview).

- **\_files:** Здесь хранятся остальные файлы (не изображения), сохраняемые CMS

системой, обычно это файлы типа – doc, pdf, txt и другие.

- **CSS:** В директории CSS находятся css файлы веб-сайта и админ-панели CMS системы.
- **data:** В этой директории хранятся файлы, сохраняемые JavaScript модулем WYSIWYG, который используется в админ-панели CMS системы.
- **img:** Здесь хранятся графические файлы, которые используются для дизайна веб-сайта, например логотип, кнопки, иконки и т.д.
- **js:** Эта директория содержит JavaScript файлы, модули веб-сайта и админ-панели CMS.

Теперь пришла очередь рассмотрению основных директорий фреймворка:

- **application:** Содержит файлы и директории веб-сайта, в том числе файлы реализующие модель проектирования MVC. В этой категории содержатся следующие подкатегории:

**cache** - Здесь хранятся кеш файлы, веб-сайта. Кеш это – файлы создаваемые системой для хранения временной информации.

**classes** – В этой директории содержатся две подкатегории MVC структуры: Controller – Файлы контроллеры и Model – файлы модели.

**config** – Эта директория предназначена для конфигурационных файлов.

**views** – Содержит файлы представления (файлы вида MVC структуры). Представления являются обычными php-файлами, которые используются для генерации HTML или другого вида вывода информации.

- **modules:** Здесь содержатся директории пользовательских модулей фреймворка, в том числе CMS система.
- **system:** Это ядро фреймворка Kohana. Содержащее в себе системные файлы и модули.

В каждой директории – `modules` и `system`, находятся модули с аналогичной, файловой структурой как в `application`. То есть у каждого модуля должны быть директории – `classes`, `views` и т.д.

Как описывалось ранее, Kohana это – HMVC (Hierarchical Model View Controller – Иерархическая Модель-Вид-Контроллер). Это означает, что файловая система фреймворка использует иерархическую структуру директорий. Иерархия в Kohana построена по следующему принципу: Когда мы хотим применить какой ни-будь модуль, то в первую очередь этот запрос поступает в `application`, если в `application` нет искомого модуля, запрос передается в `modules` и потом только в `system`. То есть если мы хотим переопределить системный модуль фреймворка, то мы его копируем в папку `modules` и там его переопределяем, тем самым не нарушая исходный системный модуль фреймворка. Благодаря этой технологии можно дополнить или модифицировать функционал фреймворка.

#### **1.4.4 Панель администрирования CMS**

У каждой CMS системы должна быть панель администрирования сайта. С помощью этой панели, пользователь может: изменять, создавать и удалять информацию сайта, не беспокоясь о его поломке. У CMS системы существует два права доступа в админ-панель это - права администратора и права пользователя. Администратору доступны все функции CMS, где он может манипулировать - таблицами в базе данных, модулями, разделами сайта и многое другое. С правом доступа пользователя, доступен ограниченный функционал – это изменение данных и создания новых разделов сайта. Перед рассмотрением административной панели, будет приведено краткое описание установки CMS.

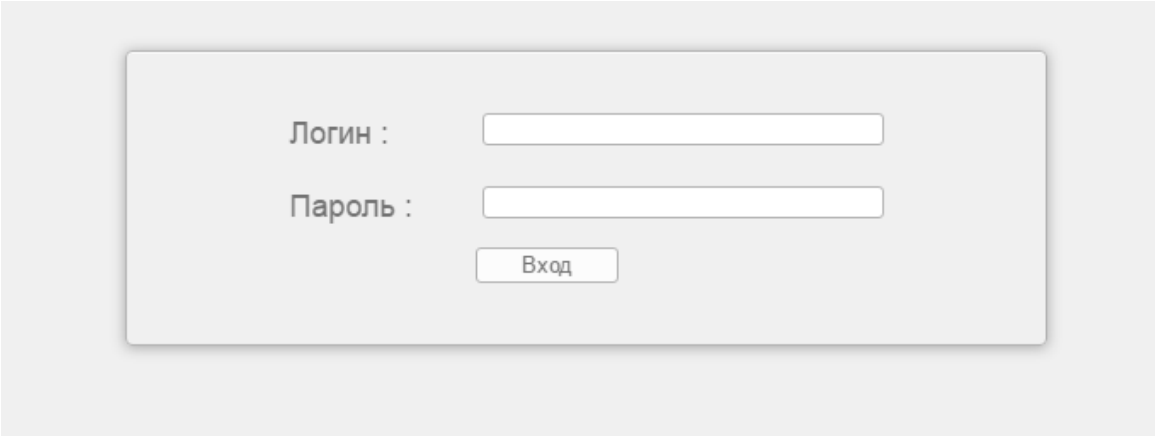


Перед установкой есть предположение, что веб-сервер подготовлен, к размещению веб-сайтов. Для того что бы установить CMS систему на веб сервер, потребуются определенные знания в веб-технологиях, я приведу только объективную последовательность действий. Файлы CMS находятся на CD диске, который прилагается к ВКР работе.

Установка CMS системы на веб-сервер происходит следующим образом:

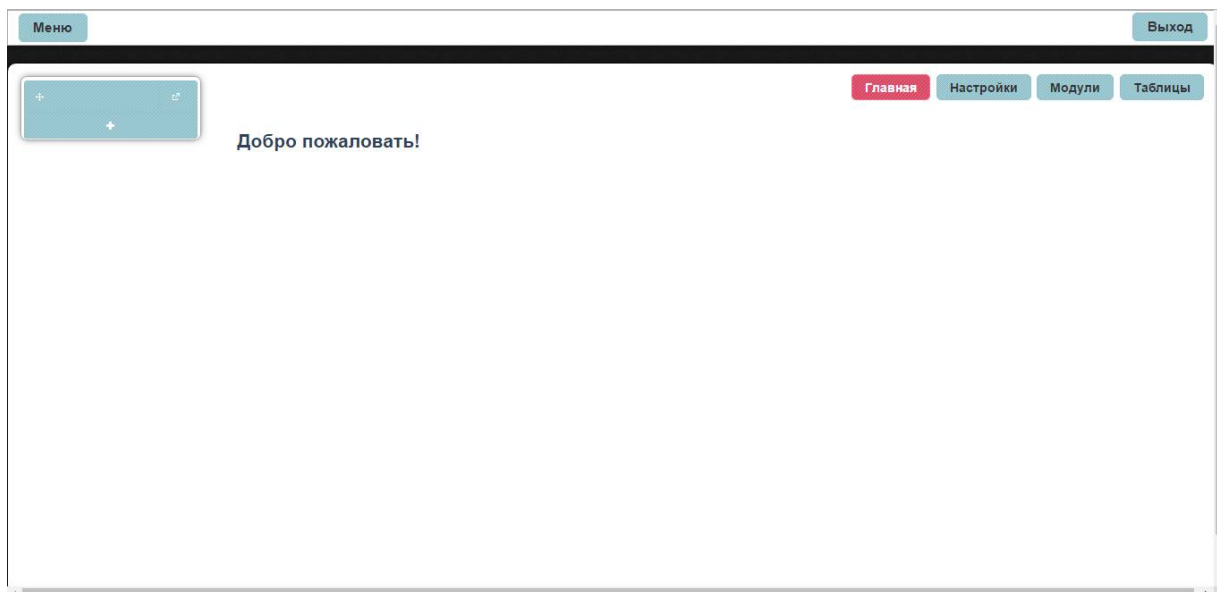
Копируются файлы CMS на веб-сервер, затем необходимо создать базу данных, разместить в нее таблицы CMS (cms.sql), затем необходимо отредактировать конфигурационные файлы фреймворка и CMS система готова к работе.

Если установка прошла успешно, админ-панель будет доступна по адресу [www.адрес сайта /admin](#), где необходимо будет ввести логин и пароль что бы войти в админ-панель (см. рис. 21). По умолчанию нужно ввести для доступа логин – root, пароль-123456.



Форма входа в админ-панель CMS системы. Рис. 21

После авторизации, открывается страница админ-панели (рис. 22) с правами администратора.

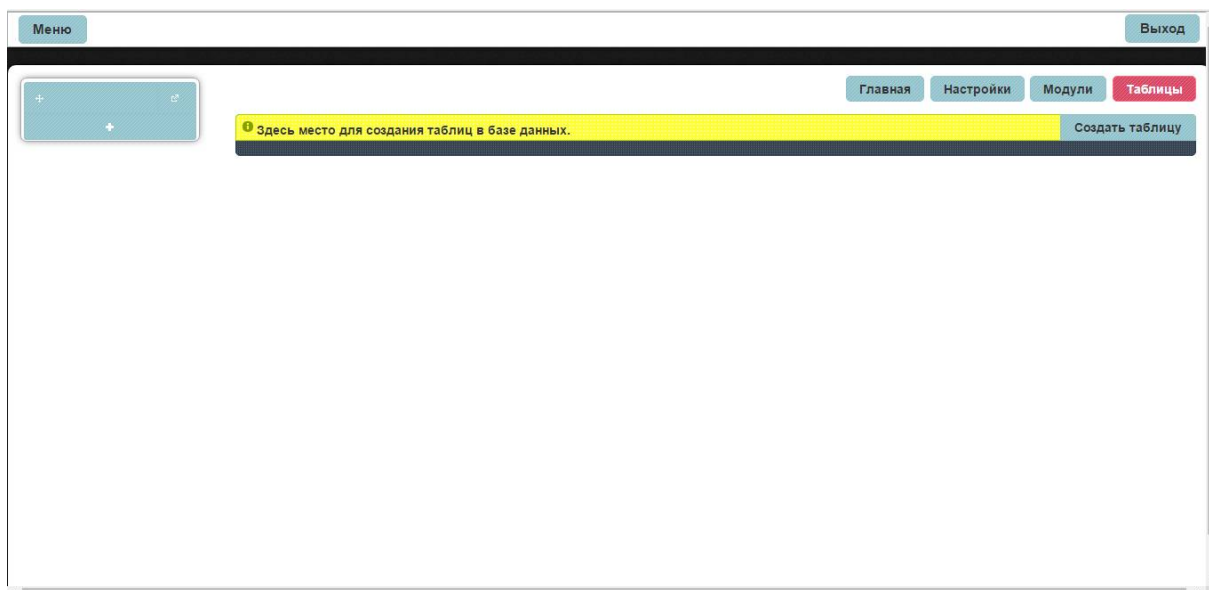


Главная страница админ-панели CMS системы. Рис. 22

На рис. 22, главная страница админ-панели CMS. На главной странице выводится приветствие - Добро пожаловать. С лева находится панель разделов сайта, которая еще не заполнена. С правой стороны размещены кнопки для перехода на технические страницы админ-панели, которые далее будут рассмотрены. В самом верху, размещены две кнопки – Выход (выход из админ-панели) и меню (для управления пользователями и изменения тем стилизации) которое будет рассмотрено позже.

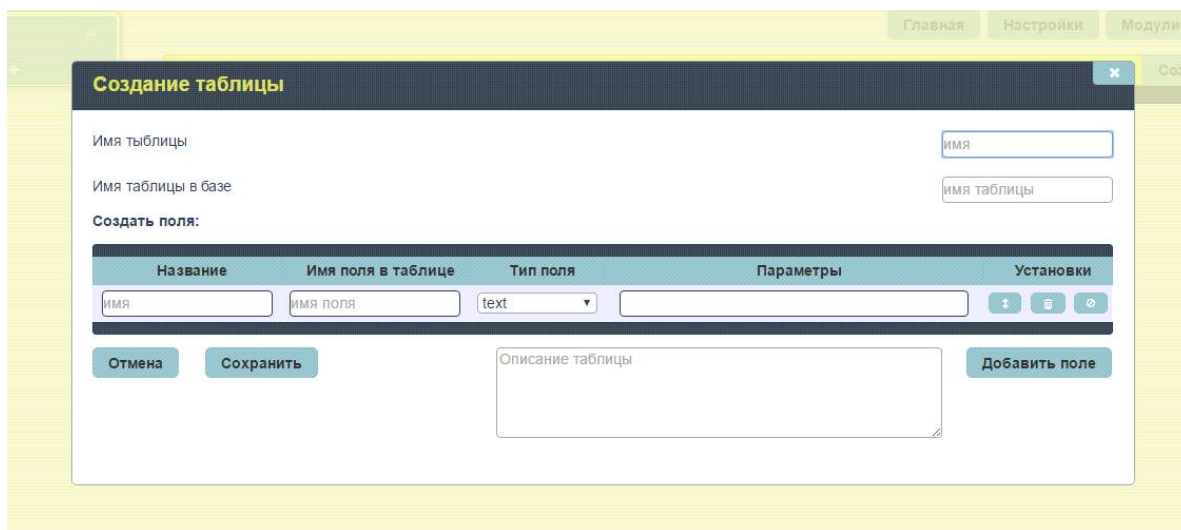
В следующих главах, для наглядного примера, будут приведены изображения административной панели CMS системы, до создания и после создания интернет-магазина «Витязь».

#### 1.4.4.1 Таблицы



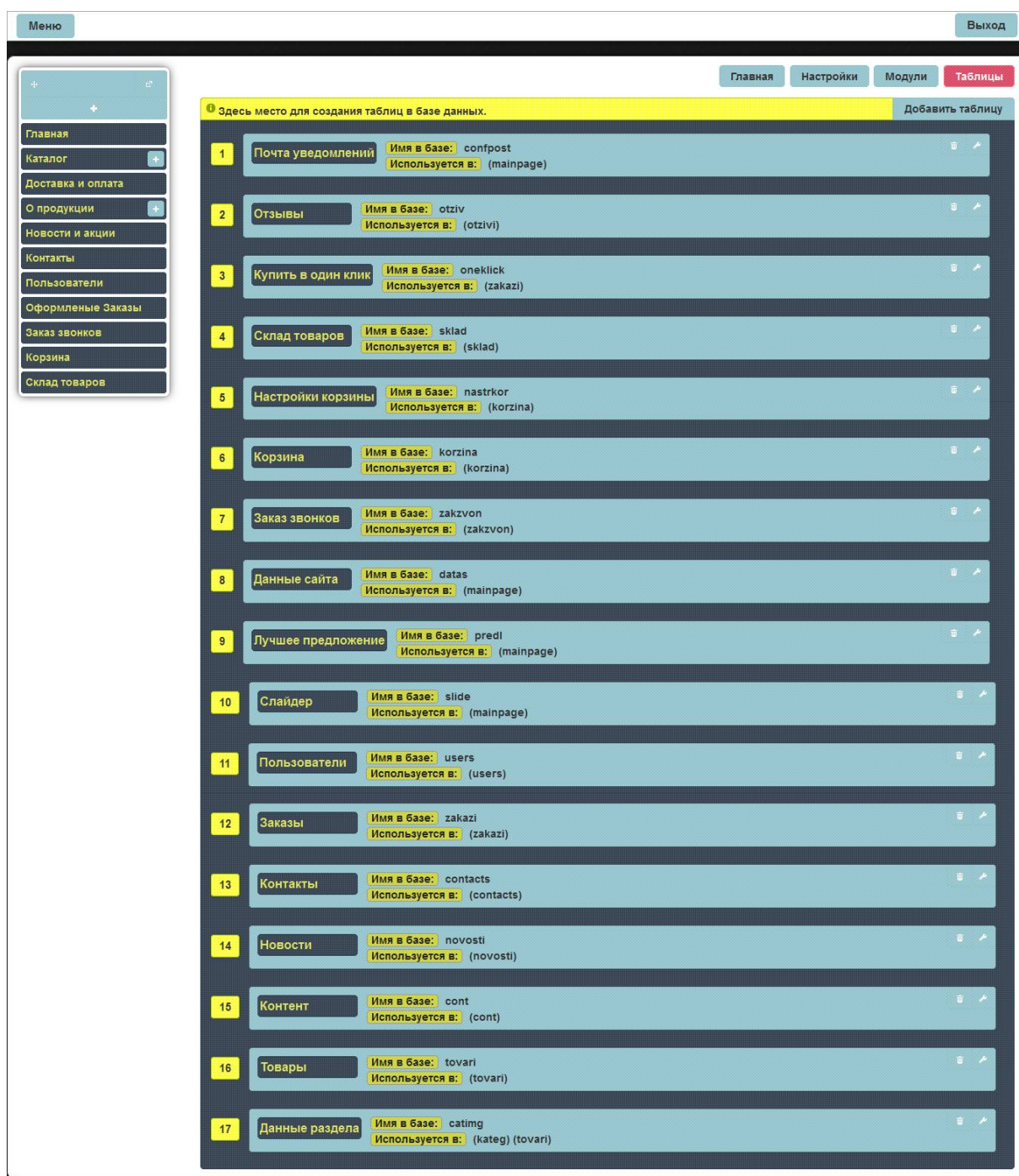
Страница создания таблиц в базе данных. Рис. 23

Этот страница (рис.23), предназначен для создания таблиц в базе данных. При нажатии на кнопку «Создать таблицу», появляется модальное окно (рис. 24), с помощью которой, создается таблица в базе данных.



Модальное окно создания таблицы в базе данных. Рис. 24

Далее изображения страницы - «Таблицы»: списка таблиц (рис.25) и модальное окно изменения таблицы (рис.26), интернет-магазина «Витязь».



Список таблиц интернет-магазина «Витязь». Рис. 25

На рис. 25, список таблиц, каждая таблица будет использована модулем — о модулях будет описано в следующей главе.

Изменение таблицы: ( tovari )

Имя таблицы

Товары

Изменить поля:

Название	Имя поля в таблице	Тип поля	Параметры	Установки
Артикул	artikul	text		↑ ↓ ✕
Заголовок	title	text		↑ ↓ ✕
Изображения	imgs	multiple img		↑ ↓ ✕
Цена	cena	text		↑ ↓ ✕
Старая цена	scena	text		↑ ↓ ✕
Новинка	nova	radio	value: Нет Да;	↑ ↓ ✕
Сезон	sezon	multiple select	value: Зима Весна Лето Осень Демисезон;	↑ ↓ ✕
Вариация товара	variac	select	value: Красная Зеленая Черная Серебрянная Голу	↑ ↓ ✕
Производитель (Назван	brendn	select	value: Alfa industris Alfa industris 1 Alfa industris 2;	↑ ↓ ✕
Производитель (фото)	brendf	img		↑ ↓ ✕
Производитель (Стран	brends	text		↑ ↓ ✕
Контент	cont	html		↑ ↓ ✕
SEO текст	seo	html		↑ ↓ ✕
URL	url	text		↑ ↓ ✕
Title (meta)	ptitle	text		↑ ↓ ✕
Description (meta)	pdesc	text		↑ ↓ ✕
Keys (meta)	pkeys	text		↑ ↓ ✕

Отмена

Сохранить

Описание таблицы

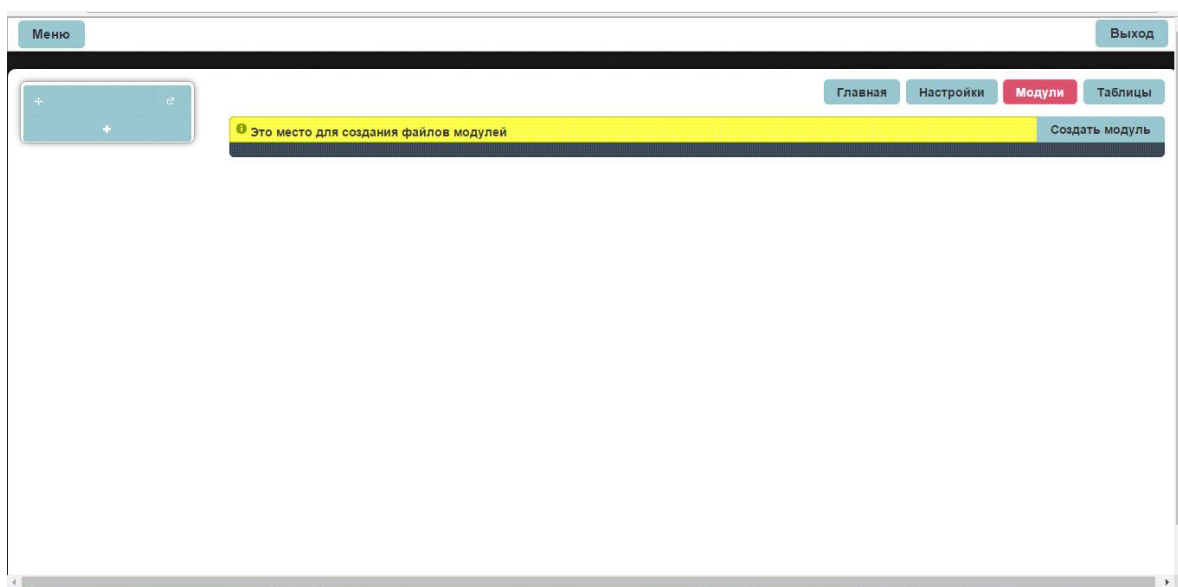
Добавить поле

Модальное окно таблицы товаров. Рис. 26.

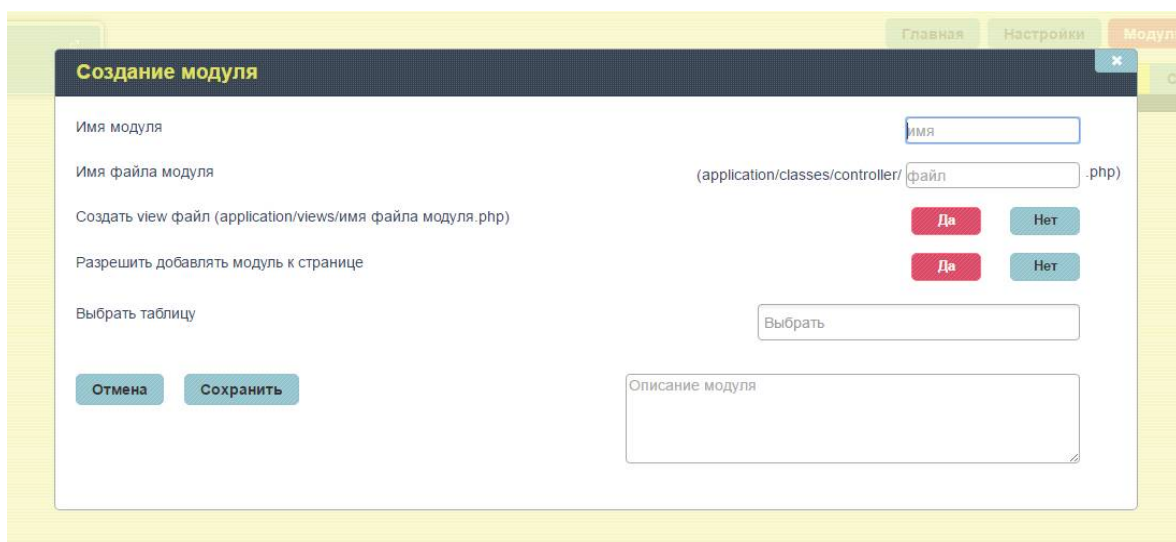
На рис. 26, одна из таблиц интернет-магазина «Витязь», в данном случае это таблица товаров. В столбце – «Название», выводятся названия столбцов – которые буду использованы в таблицах раздела сайта в админ-панели (см. [стр. 61](#)). «Имя поля в таблице» - это название столбцов таблиц в базе данных. «Тип поля» - это установка для ввода типа данных в ячейке таблицы из разделов сайта в админ-панели (см. [стр. 61](#)). «Параметры» служат для установки параметров для определенных типов полей. В столбце «Установки», находятся кнопки для сортировки, скрывтия и удаления строк.

#### 1.4.4.2 Модули

Страница – «Модули» предназначена для работы с модулями. Модуль CMS – это контроллер фреймворка, которого расширяет абстрактный класс CMS системы и тем самым дополняет его дополнительным функционалом. При создании модуля с помощью админ-панели, CMS система, следуя MVC модели, создает контроллер и «вид» в фреймворке. А «Модель» (доступ к данным), реализовано в CMS.



Страница – Модули. Рис. 27



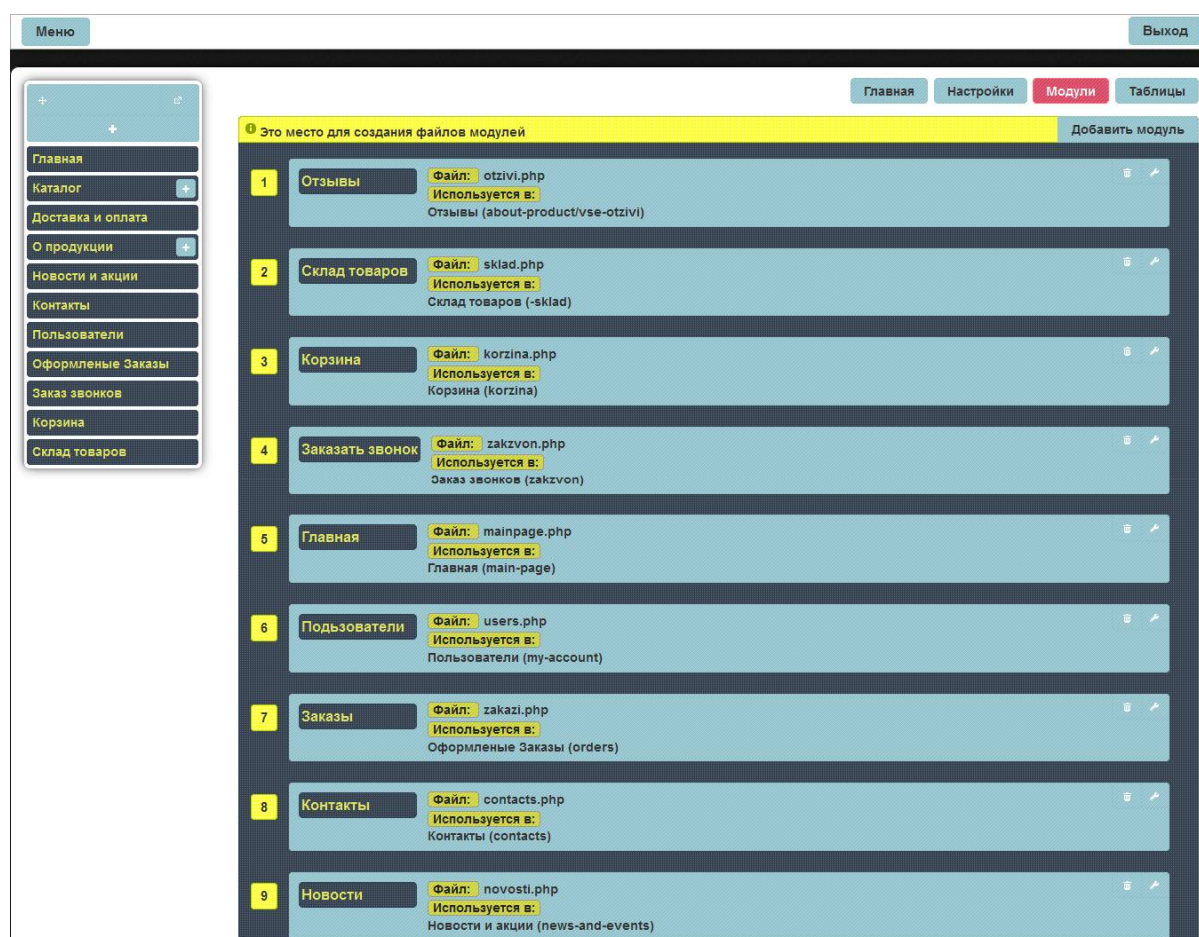
Модальное окно создания модуля. Рис. 28

На рис. 28, модальное окно создания модуля в CMS системе. Для создания модуля необходимо заполнить все необходимые поля, это – Имя

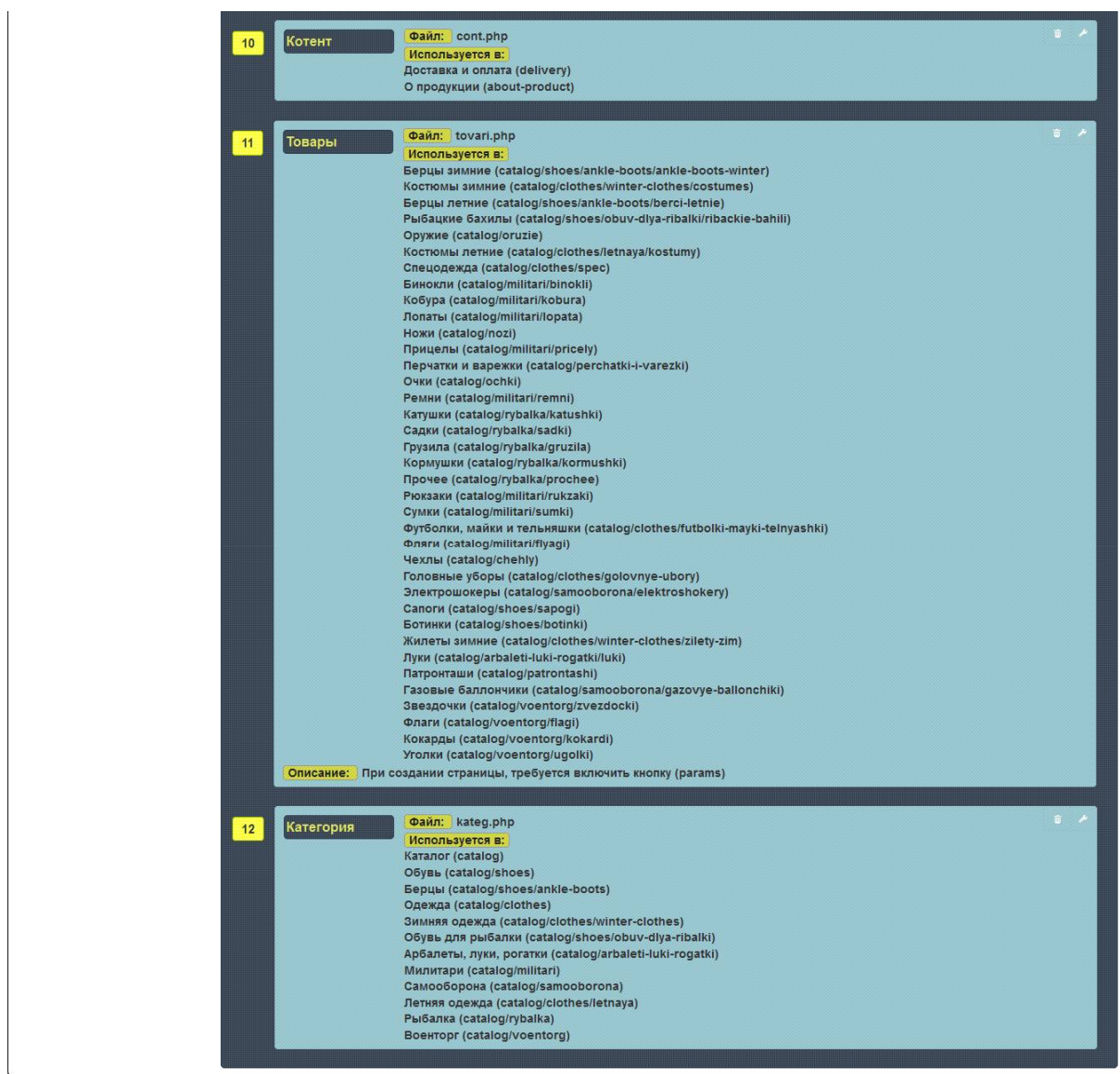


модуля, имя файла модуля (название файла и класса). Есть два переключателя, это – Создать view файл (означает создать контроллер вместе с view) и разрешить добавлять модуль к странице (это задает видимость модуля для пользователей админ-панели). А теперь самое важное поле ввода это – Выбрать таблицу. В списке для выбора таблиц выводятся таблицы, которые были ранее созданы на странице «Таблицы». Можно выбрать несколько таблиц, которые буду привязаны к этому модулю, и при написании кода в этом модуле, уже будет предоставлен интерфейс, для работы с подключенными таблицами.

Далее на рис. 29, представлено изображение страницы – «Модули», интернет-магазина «Витязь».



Продолжение изображения на след. Странице.



Раздел – Модули. Рис. 29

У каждого модуля в списке, есть дополнительная информация: это название файла модуля в файловой системе, в каком разделе CMS системы, этот модуль используется и описание. Как видно из рис. 29, самый используемый модуль это – Товары.



### 1.4.4.3 Разделы структуры CMS

Разделы CMS системы. Это древовидная структура, где каждый раздел может быть страницей сайта или просто разделом CMS, для работы с данными.

Меню

Выход

Главная Настройки Модули Таблицы

Костюмы зимние Перейти по ссылке : /catalog/clothes/winter-clothes/costumes

root >> Каталог >> Одежда >> Зимняя одежда >> Костюмы зимние

Данные раздела + Добавить запись

Показать 10 записей Поиск:

№	Изображение раздела	SEO текст	Дополнительно
1		HTML	

Записи с 1 до 1 из 1 записей

Товары + Добавить запись

Показать 10 записей Поиск:

№	Артикул	Заголовок	Изображения	Цена	Старая цена	Новинка	Дополнительно
1	Костюм зимний Метель (мембрана)	Костюм зимний Метель		4000		Нет	
2	Костюм зимний удлиненный Арктика (мембрана)	Костюм зимний удлиненный Арктика		4500		Нет	
3	Куртка зимняя удлиненная Лес (оксфорд)	Куртка зимняя удлиненная Лес		1800		Нет	
4	Куртка N-2В	Куртка N-2В		5800		Нет	
5	Куртка N-3В	Куртка N-3В		6800		Нет	
6	Костюм Парка	Костюм Парка		7800		Нет	

Записи с 1 до 6 из 6 записей

Разделы CMS. Рис. 30.

На рис.30, изображение разделов структуры CMS, интернет-магазина «Витязь». В левой части (в виде столбика разделов) находятся разделы CMS. При клике на один из разделов, в правую часть (где было приветствие – Добро пожаловать), подгружаются таблицы активного раздела. Как видно на рис.30, активным разделом является – Костюмы зимние, а данные этого раздела

являются таблицы – «Данные раздела» и «Товары». О таблицах раздела будет описано в следующей главе.

Древовидная структура разделов CMS — это та же структура веб сайта. У каждого раздела должен быть свой уникальный URL, независимо от того, будет ли он рабочим на сайте или нет. Далее рассмотрим раздел подробнее.



Активный раздел CMS системы. Рис. 31

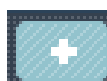
У каждого активного раздела, доступны кнопки, это –



- удаление раздела. Удаляет раздел перед подтверждением действия.



- изменение раздела. Вызывает модальное окно для изменения (рис. 32).



- создание дочернего раздела. Вызывает модальное окно для создания дочернего раздела (рис. 33).



- скрыть раздел. Скрывает раздел, изменяя тем самым цвет кнопки на красный и URL не доступен.



- переместить раздел. При активации этой кнопки, доступно перемещение раздела.

**Изменение страницы**

Название раздела сайта (или страницы): Костюмы зимние

Адресс страницы URL: /catalog/clothes/winter-clothes/ costumes

Заголовок страницы (title-мета данные страницы): Купить зимний костюм в Екатеринбурге | Интернет-магазин низких цен

Описание страницы (description-мета данные страницы): Зимние костюмы по низким ценам для любых холодов покупайте в интернет-

Ключевые слова страницы (keys-мета данные страницы): купить зимний костюм, зимние костюмы екатеринбург, купить зимний костюм

Выбрать модуль: Товары (tovari)

Включить параметры в URL: /catalog/clothes/winter-clothes/costumes/(params) (Включать если требуется модулю) **Включить**

Задать разрешения страницы на: **+ Создание** **Изменение** **Удаление**

**Отмена** **Сохранить**

Модальное окно изменения раздела. Рис.32

**Создание страницы**

Название раздела сайта (или страницы): Название

Адресс страницы URL: /catalog/clothes/winter-clothes/costumes/ адрес

Заголовок страницы (title-мета данные страницы): Заголовок

Описание страницы (description-мета данные страницы): Описание

Ключевые слова страницы (keys-мета данные страницы): Ключевые слова

Выбрать модуль: Выбрать

Включить параметры в URL: /catalog/clothes/winter-clothes/costumes/адрес/(params) (Включать если требуется модулю) **Включить**

Задать разрешения страницы на: **+ Создание** **Изменение** **Удаление**

**Отмена** **Сохранить**

Модальное окно изменения раздела. Рис.33

На рис.32 и 33, модальные окна создания и изменения раздела. У каждого раздела, должно быть - название, URL и модуль. Остальные данные это – Мета данные HTML страницы, кнопка включения параметров (если требуется разделу) – то есть get данные прилагаемые к URL раздела.

#### 1.4.4.4 Данные раздела

У каждого раздела, свои данные, где в роли хранилища служат общие таблицы базы данных. На рис.34, к разделу подключен модуль, к которому подключены две таблицы — это «Данные раздела» и «Товары».

Костюмы зимние






Перейти по ссылке : /catalog/clothes/winter-clothes/costumes

root >> Каталог >> Одежда >> Зимняя одежда >> Костюмы зимние

Данные раздела + Добавить запись

Показать 10 записей

Поиск































№	Изображение раздела	SEO текст	Дополнительно
1		HTML	   

Записи с 1 до 1 из 1 записей

Товары + Добавить запись

Показать 10 записей

Поиск

№	Артикул	Заголовок	Изображения	Цена	Старая цена	Новинка	Дополнительно
1	Костюм зимний Метель (мембрана)	Костюм зимний Метель		4000		Нет	   
2	Костюм зимний удлиненный Арктика (мембрана)	Костюм зимний удлиненный Арктика		4500		Нет	   
3	Куртка зимняя удлиненная Лес (оксфорд)	Куртка зимняя удлиненная Лес		1800		Нет	   
4	Куртка N-2B	Куртка N-2B		5800		Нет	   
5	Куртка N-3B	Куртка N-3B		6800		Нет	   
6	Костюм Парка	Костюм Парка		7800		Нет	   

Записи с 1 до 6 из 6 записей

Таблицы раздела. Рис. 34

Далее представлены изображения модальных окон создания и изменения записи в таблице раздела.







Исходя из выше приведенных взаимодействий модулей и таблиц базы данных, для лучшего понятия, я приведу последовательность действий создания рабочего раздела сайта. Все действия осуществляются через админ-панель.

Сначала создается таблица (одна или несколько). Далее создается модуль, в котором подключаются ранее созданные таблицы. После создания модуля, в файловой системе фреймворка создается файл «контроллер» и «вид». Затем создается раздел сайта, к которому подключается модуль. Далее в модуле пишется программный код. Вот и все что нужно сделать для разработки модуля.

Этот модуль может быть использован, не только одним разделом. На одном модуле могут работать от 1 до N разделов. Примером служит модуль - товары, который был уже рассмотрен ранее. Это можно ассоциировать с «полиморфизмом», где модуль это - базовый класс, а разделы производные его. Хотя это не прямая терминология этой структуры. Модуль лишь описывает общую логику, а у каждого раздела, свои данные. Надеюсь было понятно описано взаимодействие разделов с модулями в CMS системе.

### **1.4.5 Заключение**

Первая часть выпускной квалифицированной работы, была посвящена методам и технологиям, с помощью которых разработан интернет-магазин «Витязь». Так же была приведена структура интернет-магазина, которая не отличается от структуры разделов CMS системы.

Во второй части выпускной квалифицированной работы, будет рассмотрен программный код модулей, API (Application programming interface – программный интерфейс) CMS системы и дополнительные технологии которые, упрощают, и ускоряют разработку сайтов.

## Глава 2. Разработка интернет-магазина «Витязь»

Практическая часть посвящена программному коду интернет-магазину «Витязь». Будет рассмотрен каждый модуль CMS системы. Весь исходный код интернет-магазина, размещен на CD диске, который прилагает к выпускной квалифицированной работе.

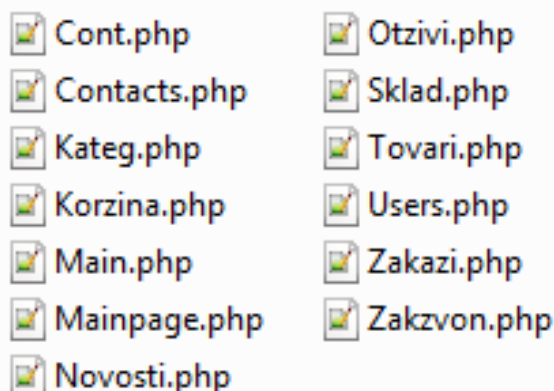
В следующих главах, будут рассмотрены файлы модулей: их расположение и количество, код модуля до его модификации, а так же каждый модуль, который отвечает за работу интернет-магазина – «Витязь»

### 2.1 Размещение файлов модулей

В CMS системе как описывалось ранее, файлы модулей создаются через панель администрирования. При создании модуля, CMS система создает файл контроллер в директории `\application\classes\Controller`, и файл вида (если требуется модулю) в `\application\views`. В этих файлах программный код интернет-магазина – «Витязь».

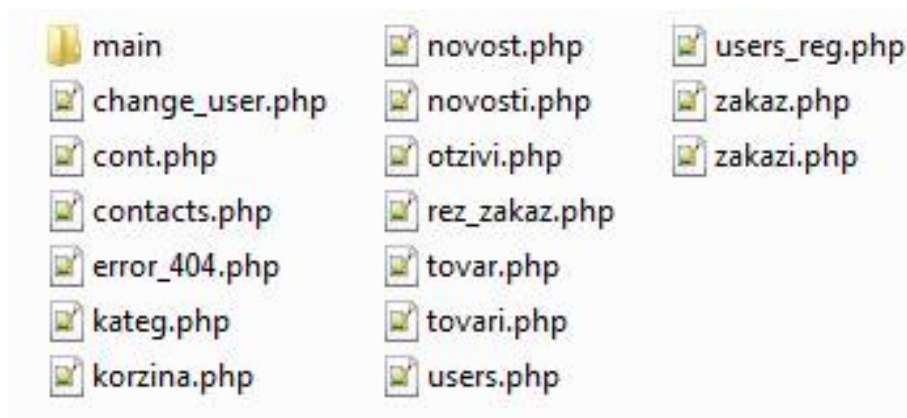
Все остальные файлы и директории, которые относятся к фреймворку, к модулям фреймворка и CMS системе, не будут рассмотрены.

Содержимое директории – `\application\classes\Controller`:

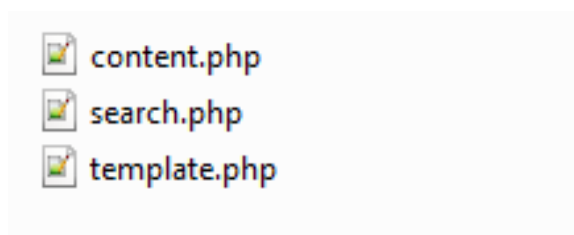




Содержимое директории – \application\views:



Содержимое директории – \application\views\main:



## 2.2 Код модуля до модификации

Далее рассмотрим код файлов модуля, до их модификации.

**Листинг 1.** Код контроллера Module.php до модификации.

```
1 <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3 class Controller_Module extends CMS_Template { //Template
```

```
4 // class Controller_Module extends CMS_Controller { //Controller
```

```
5
```

```
6
```

```
7     public function action_index()
```

```
8     {
```

```
9
```

```
10         $urlsite = URL::site();
```

```
11
```

```
12         $this->template->pid = CMS::$pid;
```

```
13         $this->template->title = CMS::$ptitle;
```

```
14         $this->template->pkeys = CMS::$pkeys;
```

```
15         $this->template->pdesc = CMS::$pdesc;
```

```
16         $this->template->crumbs = CMS::crumbs();
```

```
17         $this->template->content = Twig::factory('module')
```

```
18                                     ->set('urlsite',$urlsite)
```

```
19                                     ->set('title',CMS::$pname);
```

```
20     }
```

```
21
```

```
22
```

```
23 }
```

На листинге 1, код контроллера в изначальном его виде, то есть до модификации. Класс CMS\_Template, расширяет Controller\_Module . Свойство класса \$this->template, - это объект который содержит данные HTML шаблона.

Статические свойства и метода класса CMS будут рассмотрены следующей главе. Стоит отметить метод Twig, это шаблонизатор который отвечает за формирование вида (View) CMS системы.

**Листинг 2.** Код вида module.php до модификации.

```
<h2>{{title}}</h2>
```

Файл вида на листинге 2, содержит всего одну строчку, - это HTML тег заголовка <h2>. {{title}} - это выражение шаблонизатора Twig.

## 2.3 API CMS системы

API CMS, предоставляет свойства и методы для работы с данными. Система тем самым реализует Model MVC структуры. У каждого модуля под кодом есть закомментированный участок, описывающий API CMS системы.

Свойства, предоставляемые CMS системой:

CMS::\$pid; id страницы или id раздела

CMS::\$parid; id родителя страницы

CMS::\$pname; Название раздела в меню CMS

CMS::\$ptitle; Заголовок страницы (title)

CMS::\$pdesc; Описание страницы (meta)

CMS::\$pkeys; Ключевые слова страницы (meta)

CMS::\$tnames; массив, имен таблиц связанных с модулем.

CMS::\$level; уровень вложенности в дереве

CMS::\$uri; uri страницы без params path/to/page

Методы для работы с деревом сайта:

`CMS::crumbs($id = null);`

«Хлебные крошки» - (список разделов от родителя), возвращает ассоциативный массив name, url, id.

`CMS::depth($depth=1,$id = 1,$offset = 0,$limit = 1000,  
$only_visible = null,$table = 'a_paths');`

Возвращает глубину дерева дочерних элементов.

`CMS::branch ($id = 1, array $select = null);`

Вывод всей ветки дочерних узлов без родителя.

\$ id-родитель, \$select-перечень имен полей

`CMS::final_node ($id = 1);`

Нахождение всех конечных узлов родителя (самые крайние на ветке).

`CMS::get_parent($id) ;`

Возвращает родителя.

Методы для работы с данными;

`CMS::instance()->get_data();`

Возвращает данные связанные с разделом.

Аналог выражения:

`DB::select()->from($tnames[0])`

`->where('idpath','=',CMS::$pid)`

`->where('visible','=',1)`

`->order_by('sort')->execute()->as_array();`

```
CMS::instance()->where('name','=', 'name')->get_data();
```

Возвращает данные по условию связанные с разделом.

```
CMS::instance()->select('name','img','date')->get_data();
```

Возвращает данные по перечислению полей.

```
CMS::instance()->from('tname')->get_data();
```

Возвращает данные по имени таблицы.

```
CMS::instance()->from()
```

```
    ->where->limit()->offset()
```

```
    ->execute()->cached()->as_array();
```

(->where): объект Query Builder-а для дополнительных условий.

```
CMS::instance()->order_by('sort','desc');
```

Возвращает данные, включая сортировку.

```
CMS::values_field($fname = "");
```

Возвращает значения всех полей radio или checkbox или multipleselect, по имени поля \$fname = название поля

```
CMS::parse_tree($tree,$pid = 1);
```

Возвращает массив в виде древовидной структуры

```
CMS::instance()->insert(array('fname' => 'fdata',
```

```
    'fname' => 'fdata'),$tname = null);
```

Вставляет данные в таблицу по умолчанию в \$tnames[0].

```
CMS::instance()->update($tname = null)
                    ->set(array('username' => 'jane'))
                    ->where('username', '=', 'john')->execute();
```

Обновляет данные в таблице по умолчанию в \$tnames[0].

Методы для работы с пользователями:

В таблице должно быть два поля name и pass.

```
CMS::user_is_live($timestamp = null(deff 3600*24));
```

Проверка на окончание сессии. Если сессия не закончилась, то продлеваем сессию на \$timestamp, при удаче возвращает данные пользователя, иначе вернет false.

```
CMS::user_in(['name' => " , 'pass' => "], $table = null, $idpath = null,
              $active = true, $timestamp = null(deff 3600*24));
```

Вход пользователя. Если пользователь ранее занесен в базу данных и не скрыт, то создается сессия и возвращает true, иначе false.

```
CMS::user_exist(['name' => " , 'pass' => "], $table = null, $idpath = null);
```

Проверка на существование пользователя в базе, при удаче возвращает данные пользователя, иначе false.

```
CMS::user_name_exist($name = null, $is_visible = null,
                    $table = null, $idpath = null);
```

Простая проверка, существует ли логин пользователя в базе или нет. При удаче true, иначе false.

```
CMS::user_reg(['name' => " , 'pass' => "], $table = null, $idpath = null,  
              $active = true, $timestamp = null(deff 3600*24));
```

Регистрация. Заносит в базу и в сессию. При удаче возвращает id пользователя, только что созданной записи в таблице.

```
CMS::user_out() ; Удаляет сессию.
```

Метод сохранения изображений:

Метод сохраняет изображение в две копии, одно изображение сохраняется в папке preview с заданным ранее разрешением,

а второе как есть, оба файла сохраняются в jpg.

Список аргументов передаваемые в функцию:

\$mark = водяной знак png,

\$mark\_opa = прозрачность водяного знака,

\$scrwidth = ширина обрезки,

\$scrheight = высота обр.,

\$scr\_x = поз. x, \$scr\_y = поз. y обрезки

\$data = имя сохраняемого изображения /00/00000000/

```
CMS::setImage ($image = $_FILES['name'], $width = null, $height =  
null,
```

```
$mark = null, $mark_opa = 100, $scrwidth = null,
```

```
$scrheight = null, $scr_x = null, $scr_y = null);
```

Хелперы (Помошники):

```
CMS::group_by($key,$array);
```

Группирует массив по ключу.

```
CMS::div_array($divider,$array);
```

Делит массив на заданное количество элементов.

```
CMS::parse_date($newdate);
```

Изменяет формат даты с ууууммдд на dd месяц уууу

```
CMS::cyrillicToLatin($text, $toLowerCase = true);
```

Переводит текст с кириллицы на латиницу.

```
throw HTTP_Exception::factory(404,'Page not visible');
```

Вызов исключения, для возврата 404 ошибки (страница не найдена).

Это были все методы и функции, предоставляемые CMS системой, для работы с данными. Далее рассмотрим модули, на основе которых, функционирует интернет-магазин – «Витязь».

## 2.4 Модули интернет-магазина – Витязь

В следующих главах будет рассмотрен код модулей интернет-магазина – «Витязь». Каждая глава будет посвящена каждому модулю, где будет описан контроллер модуля, какие файлы вида используются контроллером, и какую функцию они выполняют.

## 2.4.1 Базовый контроллер– Main.php

Контроллер Main.php – использует файлы вида : template.php – это базовый шаблон, в котором HTML содержимое Header, footer, left-block и ячейка контентной области – content (см. [стр. 15](#)), где выводятся все остальные файлы вида, content.php - контент главной страницы (см. [Рис. 1.4](#)), search.php – результат поиска.

**Листинг 3.** Базовый Класс – Main.php

```
1 <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3 class Controller_Main extends Controller_Template {
```

```
4
```

```
5     public $template = 'main/template';
```

```
6
```

```
7     public function before()
```

```
8     {
```

```
9         parent::before();
```

```
10
```

```
11         $box = [];
```

```
12         $menu = '';
```

```
13         $catalog = [];
```

```
14         $userdata = '';
```

```
15         $is_live = false;
```

```
16         $main_d[0] = '';
```

```
17
```

```
18         if(!Request::initial()->is_ajax())
```

```
19         {
```

```
20             ////////////////навигация сайта
```

```
21             $menu = CMS::depth(1,1,1,5,true);
```

```
22             foreach($menu as $k => $val)
```

```
23             {
```

```
24                 if($val['id'] != 2)
```

```
25                 {
```

```
26
```

```
27                     $ch = CMS::depth(1,$val['id'],0,16,true);
```

```
28
```

```
29                     if($ch) $ch = CMS::div_array(8,$ch);
```

```
30
```

```
31                     $menu[$k]['childs'] = $ch;
```

```
32
```

```
33                 }
```

```
34             }
```



Продолжение листинга на след странице.

```
35 //////////////////////////////////////////////////каталог сайта
36 $catalog = CMS::depth(1,2,0,100,true);
37
38 //////////////////////////////////////////////////данные главной
39 $main_d = DB::select('tel','addr','email','logo')
40           ->from('datas')->execute()->cached()->as_array();
41
42 //////////////////////////////////////////////////адреса в футоре
43 $contacts = DB::select('title','subtitle','addr','tel')
44           ->from('contacts')->execute()->cached()->as_array();
45
46 $box = [];
47 $buff = $contacts;
48
49 //////////////////////////////////////////////////группируем по два адреса на один город
50 foreach($contacts as $k => $v)
51 {
52     $i = 1;
53
54     foreach($buff as $k1 => $v1)
55     {
56
57         if($v['title'] == $v1['title'] and $i <= 2)
58         {
59             $box[$k][] = $buff[$k1];
60
61             unset($buff[$k1]);
62
63             $i++;
64         }
65     }
66 }
67
68
69
```

```
70 $is_live = CMS::user_is_live();
71
72 //////////////////////////////////////////////////ОТЗЫВЫ
73 $otz = DB::select()->from('otziv')->where('visible','=',1)
74       ->order_by('sort')->limit(3)->execute()->as_array();
75
76 foreach($otz as $k => $val)
77 {
78     $d = explode(' ', $val['date']);
79
80     $otz[$k]['date'] = CMS::parse_date($d[0]);
81
82 }
83 $this->template->uri = $this->request->uri();
84 $this->template->pid = '';
85 $this->template->title = '';
86 $this->template->pkeys = '';
87 $this->template->pdesc = '';
```

Продолжение листинга на след странице.

```

88     $this->template->zakaz = Session::instance()->get_once('zakaz');
89     $this->template->urlsite = URL::site();
90     $this->template->content = '';
91     $this->template->menu = $menu;
92     $this->template->menu2 = $catalog;
93     $this->template->catalog = CMS::div_array(8,$catalog);
94     $this->template->main_data = $main_d[0];
95     $this->template->contacts = array_reverse($box);
96     $this->template->userdata = $is_live;
97     $this->template->korz = Session::instance()->get('korzina_result');
98     $this->template->is_guest = $is_live?true:false;
99     $this->template->otz = $otz;
100
101 }
102
103 }
104
105

```

```

110 public function action_index()
111 {
112
113     $urlsite = URL::site();
114     $search = $this->request->query('search');
115
116     $str = preg_replace("/([^\-a-zA-Za-яA-Я0-9ёЁ])/u", " ", $search);
117
118     if($str)//////////поиск
119     {
120
121         $result = [];
122
123         //////////поиск по товарам
124         $sub = DB::select(['url' , 'purl'], 'id')->from('a_paths');
125
126         $db = DB::select('tovari.*', 'a_paths.purl')->from('tovari')
127             ->join(array($sub, 'a_paths'), 'INNER')
128             ->on('a_paths.id', '=', 'tovari.idpath')
129             ->where('visible', '=', 1)
130             ->and_where_open();
131
132         $db->or_where('tovari.artikul', 'LIKE', '%'.trim($str).'%')
133             ->or_where('tovari.title', 'LIKE', '%'.trim($str).'%')
134             ->or_where('tovari.artikul', 'LIKE', '%'.trim($str).'%')
135             ->or_where('tovari.brends', 'LIKE', '%'.trim($str).'%')
136             ->or_where('tovari.cont', 'LIKE', '%'.trim($str).'%')
137             ->or_where('tovari.seo', 'LIKE', '%'.trim($str).'%')
138             ->or_where('tovari.sezon', 'LIKE', '%'.trim($str).'%')
139             ->or_where('tovari.brendn', 'LIKE', '%'.trim($str).'%');
140
141         $result['tovari'] = $db->and_where_close()->execute()->as_array();

```

Продолжение листинга на след странице.

```

143
144 //////////////////////////////////////////////////даннные о товаре со склада
145 foreach($result['tovari'] as $k => $v)
146 {
147     $result['tovari'][$k]['sklad'] = DB::select('ostatok','razmer')
148     ->from('sklad')
149     ->where('visible','=',1)
150     ->where('artikul','=', $v['artikul'])
151     ->cached()->execute()->as_array();
152 }
153
154 $result['tovari'] = CMS::div_array(3,$result['tovari']);
155
156 //////////////////////////////////////////////////поиск по контенту
157 $sub = DB::select('url' , 'id', 'name')->from('a_paths');
158
159 $db = DB::select('a_paths.url', 'a_paths.name')->from('cont')
160 ->join(array($sub, 'a_paths'), 'INNER')
161 ->on('a_paths.id', '=', 'cont.idpath')
162 ->where('visible','=',1)
163 ->and_where_open();
164
165 $db->or_where('cont.cont', 'LIKE', '%'.trim($str).'%');
166
167 $result['links'][] = $db->and_where_close()->execute()->as_array();
168

```

```

169
170 //////////////////////////////////////////////////поиск по новостям
171 $sub = DB::select('url' , 'id', 'name')->from('a_paths');
172
173 $db = DB::select('a_paths.url', 'a_paths.name')->from('novosti')
174 ->join(array($sub, 'a_paths'), 'INNER')
175 ->on('a_paths.id', '=', 'novosti.idpath')
176 ->where('visible','=',1)
177 ->and_where_open();
178
179 $db ->or_where('novosti.title', 'LIKE', '%'.trim($str).'%')
180 ->or_where('novosti.desc', 'LIKE', '%'.trim($str).'%')
181 ->or_where('novosti.cont', 'LIKE', '%'.trim($str).'%');
182
183
184 $result['links'][] = $db->and_where_close()->execute()->as_array();
185
186
187 //////////////////////////////////////////////////поиск по ссылкам
188 $db = DB::select('url' , 'id', 'name')->from('a_paths')
189 ->where('visible','=',1)
190 ->where('id', '<>', 15)
191 ->where('id', '<>', 14)
192 ->where('id', '<>', 13)
193 ->where('id', '<>', 25)
194 ->where('id', '<>', 26)
195 ->where('id', '<>', 27)
196 ->where('id', '<>', 54)
197 ->where('name', 'LIKE', '%'.trim($str).'%');
198
199
200 $result['links'][] = $db->execute()->as_array();
201

```

Продолжение листинга на след странице.

```
203
204
205     $this->template->content = Twig::factory('main/search')
206         ->set('search',$search)
207         ->set('result',$result)
208         ->set('urlsite',$urlsite);
209 }
210 else
211 {
212
213     $pdata = DB::select()->from('a_paths')->where('visible','=',1)
214         ->where('id','=',15)->execute()->as_array();
215
216     $this->template->pid = 1;
217     $this->template->title = $pdata[0]['title'];
218     $this->template->pkeys = $pdata[0]['pkeys'];
219     $this->template->pdesc = $pdata[0]['pdesc'];
220     ///////////////слайдер на главной
221     $slide = DB::select('foto','cont')->from('slide')
222         ->where('visible','=',1)->execute()->cached()->as_array();
223
224     ///////////////лучшие предложения
225     $predl = DB::select('art')->from('predl')
226         ->where('visible','=',1)->order_by('sort')
227         ->execute()->cached()->as_array();
228
229     if(!empty($predl))
230     {
231         $predl = $this->predl_parse($predl);//////////находим товары по артикулу
232     }
233
```

```
237
238     ///////////////Обновление каталога
239     $sub = DB::select()->from('tovari')->where('tovari.visible','=',1)
240         ->order_by('id','desc')->limit(4);
241
242     $obnova = DB::select('tovari.*','a_paths.url',array('a_paths.url', 'purl')
243         ,array('tovari.url', 'url'))
244         ->from('a_paths')->where('a_paths.visible','=',1)
245         ->join(array($sub, 'tovari'), 'INNER')
246         ->on('a_paths.id', '=', 'tovari.idpath')
247         ->cached()->execute()->as_array();
248
249     foreach($obnova as $k => $v)//////////данные о товаре со склада
250     {
251         $obnova[$k]['sklad'] = DB::select('ostatok','razmer')
252             ->from('sklad')->where('visible','=',1)
253             ->where('artikul','=', $v['artikul'])
254             ->execute()->as_array();
255     }
256
257     $obnova = CMS::div_array (2,$obnova);
258
259     ///////////////Новости
260     $sub = DB::select()->from('novosti')
261         ->where('novosti.visible','=',1)
262         ->order_by('sort')->limit(3);
```

Продолжение листинга на след странице.

```

263
264     $news = DB::select('novosti.*', 'a_paths.url', array('a_paths.url', 'purl')
265                      , array('novosti.url', 'url'))
266           ->from('a_paths')->where('a_paths.visible', '=', 1)
267           ->join(array($sub, 'novosti'), 'INNER')
268           ->on('a_paths.id', '=', 'novosti.idpath')
269           ->cached()->execute()->as_array();
270
271     foreach($news as $k => $val)
272     {
273         $news[$k]['date'] = CMS::parse_date($val['date']);
274     }
275
276
277     ///////////////////////////////////seo текст на главной
278     $text = DB::select('cont')->from('datas')->execute()->cached()->as_array();
279
280     $this->template->content = Twig::factory('main/content')
281                                ->set('slide', $slide)
282                                ->set('tops', $predl)
283                                ->set('obnova', $obnova)
284                                ->set('news', $news)
285                                ->set('text', $text)
286                                ->set('urlsite', $urlsite);
287
288 }
289
290
291 }

```

```

293 public function predl_parse ( $predl )
294 {
295
296     $db = DB::select()->from('tovari')->where('tovari.visible', '=', 1)
297           ->and_where_open();
298
299     foreach($predl as $k => $val)
300     {
301         $db->or_where('artikul', '=', $val['art']);
302     }
303
304     $sub = $db->and_where_close();
305
306     $buff = DB::select('tovari.*', 'a_paths.url', array('a_paths.url', 'purl')
307                      , array('tovari.url', 'url'))
308           ->from('a_paths')->where('a_paths.visible', '=', 1)
309           ->join(array($sub, 'tovari'), 'INNER')
310           ->on('a_paths.id', '=', 'tovari.idpath')
311           ->cached()->execute()->as_array();
312
313     $buff2 = [];

```

Продолжение листинга на след странице.

```

314 foreach($predl as $k => $val)
315 {
316     foreach($buff as $k1 => $val1)
317     {
318         if($val['art'] == $val1['artikul'])
319         {
320             $buff2[] = $buff[$k1];
321         }
322     }
323 }
324 }
325
326 foreach($buff2 as $k => $v)//////////////////////////данные о товаре со склада
327 {
328     $buff2[$k]['sklad'] = DB::select('ostatok','razmer')
329                             ->from('sklad')->where('visible','=',1)
330                             ->where('artikul','=',$v['artikul'])
331                             ->execute()->as_array();
332 }
333
334
335 return $buff2;
336
337 }
338
339

```

```

343 ////////////////////////////qapcha
344 public function action_set_qap()
345 {
346
347     $aResponse['error'] = false;
348     if(isset($_POST['action']) && isset($_POST['qaptcha_key']))
349     {
350         $qap = false;
351
352         if(htmlentities($_POST['action'], ENT_QUOTES, 'UTF-8') == 'qaptcha')
353         {
354             $qap = $_POST['qaptcha_key'];
355             echo json_encode($aResponse);
356         }
357         else
358         {
359             $aResponse['error'] = true;
360             echo json_encode($aResponse);
361         }
362     }
363     else
364     {
365         $aResponse['error'] = true;
366         echo json_encode($aResponse);
367     }
368
369     Session::instance()->set('qaptcha_key',$qap);
370     exit;
371 }
372

```

Продолжение листинга на след странице.

```
375
376     /**
377      * Not Found.
378      */
379     public function action_404()
380     {
381
382         $this->template->content = Twig::factory('error_404');
383
384     }
385
386     /**
387      * Not authorized
388      *
389      * You have to authenticate to access this resource.
390      */
391     public function action_401()
392     {
393
394     }
395
396     /**
397      * Forbidden
398      *
399      * Authentication will not change anything.
400      */
401     public function action_403()
402     {
403
404     }
405
406 }
407
```

На листинге 3, с 7 по 103 строке, код метода before. Этот метод выполняется в первую очередь, то есть перед выполнением каждого модуля CMS системы метод before, подготавливает шаблон сайта. С 110 по 291 строку, код метода action\_index, который заполняет данными контентную область – Content - главной страницы сайта и обрабатывает поисковые запросы. С 344 по 372 код функции action\_set\_qar – эта функция используется для разблокировки форм перед отправкой на сервер. С 379 по 384 функция action\_404 – эта функция выполняется, когда URL не найден, то есть если страница не найдена.



## 2.4.2 Контроллер каталога - Kateg.php

Контроллер Kateg.php – реализует структуру каталога первого и второго уровня (см. стр. [19](#) и [20](#)). Использует файл вида kateg.php.

**Листинг 4.** Контроллер каталога – Kateg.php

```
1 <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3 class Controller_Kateg extends CMS_Template { //Template
```

```
4 // class Controller_Kateg extends CMS_Controller { //Controller
```

```
5
```

```
6
```

```
7     public function action_index()
```

```
8     {
```

```
9
```

```
10         $urlsite = URL::site();
```

```
11         $crumbs = CMS::crumbs();
```

```
12
```

```
13         /////строим массив в виде дерева и проставляем ветку в (актив)
```

```
14         /////от родителя, $razdeli = массив разделов
```

```
15         $menu = self::get_menu($crumbs,$razdeli);
```

```
16
```

```
17         $childs_el = [];
```

```
18
```

```
19         $i = 0;
```

```
20         $k1 = 0;
```

```
21
```



```

22      /////находим детей текущего узла,и заполняем
23      /////$chlds_el данными каждого потомка и делим по три штуки
24      foreach($razdeli as $k => $v)    {
25
26          if($v['pid'] == CMS::$pid)
27          {
28              if($i%3 == 0) $k1++;
29
30              $chlds_el[$k1][$k]['name'] = $v['name'];
31              $chlds_el[$k1][$k]['url'] = $v['url'];
32              $chlds_el[$k1][$k]['foto'] = DB::select('foto')
33                                      ->from('cating')
34                                      ->where('idpath','=', $v['id'])
35                                      ->where('visible','=',1)
36                                      ->order_by('sort')->limit(1)
37                                      ->execute()->get('foto');
38
39              $i++;
40          }
41      }
42  }

```

Продолжение листинга на след странице.

```

43
44      $data = CMS::instance()->get_data();
45
46      $this->template->pid = 2;
47      $this->template->title = CMS::$ptitle;
48      $this->template->pkeys = CMS::$pkeys;
49      $this->template->pdesc = CMS::$pdesc;
50      $this->template->menu2 = $menu;
51
52      $this->template->content = Twig::factory('kateg')
53                          ->set('chlds',$chlds_el)
54                          ->set('data',$data)
55                          ->set('crumbs',$crumbs)
56                          ->set('urlsite',$urlsite)
57                          ->set('title',CMS::$pname);
58  }

```

```

62     public static function get_menu ($crumbs,&$razdeli = [])
63     {
64         $razdeli = CMS::depth(CMS::$level,2);
65
66         foreach($razdeli as $k => $v)
67         {
68             foreach($crumbs as $k1 => $v1)
69             {
70                 if($v['id'] == $v1['id'])
71                 {
72                     $razdeli[$k]['active'] = true;
73                 }
74             }
75         }
76
77         return CMS::parse_tree($razdeli,2);
78     }
79 }
80
81
82
83
84
85
86
87
88

```

### 2.4.3 Контроллер карточки товара – Товари.php

Контроллер – Товари.php, реализует страницы каталога третьего уровня и карточки товара (стр. [21](#) и [22](#)). Использует файлы вида – товари.php и товар.php.

**Листинг 5.** Контроллер карточки товара – Товари.php

```

1  <?php defined('SYSPATH') or die('No direct script access.');
```

---

```

2
3  class Controller_Tovari extends CMS_Template {    //Template
4  // class Controller_Tovari extends CMS_Controller {    //Controller
5
6
7      private static $_cena = '';
8      private static $_brend = '';
9      private static $_sezon = '';
10
11     public function action_index()
12     {
13
14         $url = $this->request->param('param');
15
16         $crumbs = CMS::crumbs();
17
18         if($url)
19         {
20
21
22             $tovar = CMS::instance()->from(CMS::$tnames[1])
23                                     ->where->where('url','=',$url)
24                                     ->cached()->execute()->as_array();
25
26             if(!isset($tovar[0]))
27                 throw HTTP_Exception::factory(404,"Page not find");
28
29             ////////////данные о товаре со склада
30             $tovar[0]['sklad'] = DB::select('ostatok','razmer')
31                                     ->from('sklad')->where('visible','=','1')
32                                     ->where('artikul','=',$tovar[0]['artikul'])
33                                     ->cached()->execute()->as_array();
34
35             $this->template->title = $tovar[0]['ptitle'];
36             $this->template->pkeys = $tovar[0]['pkeys'];
37             $this->template->pdesc = $tovar[0]['pdesc'];
38             $this->template->menu2 = Controller_Kateg::get_menu($crumbs);
39
40             $crumbs[['name']] = $tovar[0]['title'];
41
42             $pbuff = DB::select()->from(CMS::$tnames[1])
43                                     ->where('visible','=','1')
44                                     ->where('idpath','=','CMS::$pid')
45                                     ->where('id','<>',$tovar[0]['id'])
46                                     ->cached()->execute()->as_array();

```

Продолжение листинга на след странице.

```

50 shuffle($pbuff);
51
52 $ptovari = [];
53
54 if(isset($pbuff[0]))
55     $ptovari[] = $pbuff[0];
56
57
58 if(isset($pbuff[1]))
59     $ptovari[] = $pbuff[1];
60
61
62 if(isset($pbuff[2]))
63     $ptovari[] = $pbuff[2];
64
65
66
67 foreach($ptovari as $k => $v)
68 {
69     $ptovari[$k]['sklad'] = DB::select('ostatok','razmer')
70                                     ->from('sklad')->where('visible','=',1)
71                                     ->where('artikul','=', $v['artikul'])
72                                     ->cached()->execute()->as_array();
73 }
74
75 //////////////////////////////////////////////////вариации товара
76 $variac = DB::select('variac','url')->from(CMS::$tnames[1])
77                                     ->where('idpath','=',CMS::$spid)
78                                     ->where('visible','=', 1)
79                                     ->where('title','=', $stovar[0]['title'])
80                                     ->cached()->execute()->as_array();
81
82
83 $this->template->content = Twig::factory('tovar')
84                                     ->set('crumbs', $scrums)
85                                     ->set('tovar', $stovar)
86                                     ->set('ptovari', $ptovari)
87                                     ->set('uri', CMS::$uri)
88                                     ->set('variac', $variac)
89                                     ->set('title', $stovar[0]['title']);
90
91 }
92 else
93 {
94     $this->tovari($scrums);
95 }
96
97
98 }

```

Продолжение листинга на след странице.

```

102
103 public function tovari ($ crumbs)
104 {
105
106     $urlsite = URL::site();
107
108     $this->template->pid = 2;
109     $this->template->title = CMS::$ptitle;
110     $this->template->pkeys = CMS::$pkeys;
111     $this->template->pdesc = CMS::$pdesc;
112     $this->template->menu2 = Controller_Kateg::get_menu($ crumbs);
113
114     $cena = $this->request->query('cena');
115     $brend = $this->request->query('brand');
116     $sezon = $this->request->query('sezon');
117     $page = $this->request->query('page');
118
119     ///////////////////////////////////////////////////список производителей
120     $brends = CMS::instance()->values_field('brendn');
121     ///////////////////////////////////////////////////список сезонов
122     $sezoni = CMS::instance()->values_field('sezon');
123
124     $data = [];
125
126     ///////////////////////////////////////////////////инициализация пагинатора
127     $curr_page = 1;
128
129     $offset = 0;
130     $limit = 9;

```

```

133
134 if(preg_match("/^[\\d]{1,3}$/", $page))
135 {
136     $offset = $limit * ($page-1);
137     $curr_page = $page;
138 }
139
140 ///////////////////////////////////////////////////
141
142 $count = [];
143
144 if($cena or $brend or $sezon)
145 {
146     ///////////////////////////////////////////////////сортировка , выборка товаров по get данным
147     $data = self::filter_tovari($cena, $brend, $sezon, $sezoni,
148                               $brends, $limit, $offset);
149
150     ///////////////////////////////////////////////////данные о товаре со склада
151     foreach($data as $k => $v)
152     {
153         $data[$k]['sklad'] = DB::select('ostatok','razmer')
154                               ->from('sklad')->where('visible','=',1)
155                               ->where('artikul','=', $v['artikul'])->cached()
156                               ->execute()->as_array();
157     }
158 }

```

Продолжение листинга на след странице.

```

160 //////////////////////////////////////////////////для подсчета страниц
161 $db = DB::select('brendn','sezon')->from(CMS::$tnames[1])
162                                     ->where('visible','=',1)
163                                     ->where('idpath','=',CMS::$pid);
164
165 //////////////////////////////////////////////////проверка по сезону
166 if(in_array($sezon,$sezoni))
167 {
168     $db->and_where('sezon','=', $sezon);
169
170 }
171 //////////////////////////////////////////////////проверка по бренду
172 if(in_array($brend,$brends))
173 {
174     $db->and_where('brendn','=', $brend);
175 }
176
177 $count = $db->cached()->execute()->count();
178 //////////////////////////////////////////////////
179 }
180 else
181 {
182
183     $data = DB::select()->from(CMS::$tnames[1])->where('visible','=',1)
184                                     ->where('idpath','=',CMS::$pid)
185                                     ->limit($limit)->offset($offset)
186                                     ->order_by('sort')->cached()
187                                     ->execute()->as_array();
188
189 //////////////////////////////////////////////////данные о товаре со склада
190 foreach($data as $k => $v)
191 {
192     $data[$k]['sklad'] = DB::select('ostatok','razmer')
193                                     ->from('sklad')->where('visible','=',1)
194                                     ->where('artikul','=', $v['artikul'])
195                                     ->cached()->execute()->as_array();
196 }
197
198 //////////////////////////////////////////////////для подсчета страниц
199 $count = DB::select('brendn','sezon')->from(CMS::$tnames[1])
200                                     ->where('visible','=',1)
201                                     ->where('idpath','=',CMS::$pid)
202                                     ->cached()->execute()->count();
203 //////////////////////////////////////////////////
204 }
205
206
207 //////////////////////////////////////////////////подсчитаем сколько страниц
208 $pages = ceil($count/$limit);
209
210
211 //////////////////////////////////////////////////отфильтруем по производителю и по сезону,
212 //////////////////////////////////////////////////специально для фильтра на сайте
213 $to_filtr = DB::select('brendn','sezon')->from(CMS::$tnames[1])
214                                     ->where('visible','=',1)
215                                     ->where('idpath','=',CMS::$pid)
216                                     ->cached()->execute()->as_array();
217
218 $list_br = self::filtr_brends($brends,$to_filtr);
219 $list_se = self::filtr_sezon($sezoni,$to_filtr);
220 //////////////////////////////////////////////////

```

Продолжение листинга на след странице.

```

223 //////////////// делим по три штуки
224 $tovari = [];
225 $i = 0;
226 $k1 = 0;
227
228 foreach($data as $k => $v)
229 {
230     if($i%3 == 0) $k1++;
231
232     $tovari[$k1][$k] = $data[$k];
233
234     $i++;
235 }
236
237 ////////////////
238
239 $cont = CMS::instance()->get_data();
240
241 $this->template->content = Twig::factory('tovari')
242     ->set('urlsite', $urlsite)
243     ->set('crumbs', $crumbs)
244     ->set('tovari', $tovari)
245     ->set('uri', CMS::$uri)
246     ->set('cont', $cont)
247     ->set('pages', $pages)
248     ->set('list_br', $list_br)
249     ->set('list_se', $list_se)
250     ->set('f_cena', self::$_cena)
251     ->set('f_brand', self::$_brend)
252     ->set('f_sezon', self::$_sezon)
253     ->set('curr_page', $curr_page)
254     ->set('title', CMS::$pname);
255
256 }

```

```

260 public static function filter_tovari ($cena, $brend, $sezon, $sezoni,
261                                     $brends, $limit, $offset)
262 {
263     $data = [];
264
265     $db = DB::select()->from(CMS::$tnames[1])
266         ->where('visible', '=', 1)
267         ->where('idpath', '=', CMS::$pid);
268
269     $flag = false;
270
271     $exec_order = false;
272     ////////////////выборка по сезону
273     if(in_array($sezon, $sezoni))
274     {
275         $db->and_where('sezon', 'LIKE', '%'.$sezon.'%');
276
277         self::$_sezon = $sezon;
278
279         $flag = true;
280     }

```

Продолжение листинга на след странице.



```

282
283 //////////////////////////////////////////////////выборка по бренду
284 if(in_array($brend,$brends))
285 {
286     $db->and_where('brendn','=', $brend);
287
288     self::$_brend = $brend;
289
290     $flag = true;
291 }
292 //////////////////////////////////////////////////цена по возрастанию
293 if($cena == 'desheвле')
294 {
295     $db->order_by('cena');
296
297     $flag = true;
298
299     self::$_cena = 'desheвле';
300
301     $exec_order = true;
302
303 }
304 //////////////////////////////////////////////////цена по убыванию
305 if($cena == 'doroже')
306 {
307     $db->order_by('cena','desc');
308
309     $flag = true;
310
311     self::$_cena = 'doroже';
312
313     $exec_order = true;
314
315 }
316
317
318
319 if($flag)
320 {
321
322     if(!$exec_order) $db->order_by('sort');
323
324     $data = $db->limit($limit)->offset($offset)
325         ->cached()->execute()
326         ->as_array();
327
328 }
329 else throw HTTP_Exception::factory(404,"Page not visible");
330
331
332 return $data;
333
334
335 }

```

Продолжение листинга на след странице.



```

338 public static function filtr_brends($brends,$count)
339 {
340
341     $box = [];
342
343     foreach($brends as $k => $v)
344     {
345         foreach($count as $k1 => $v1)
346         {
347             if($v == $v1['brendn'])
348             {
349                 $box[] = $v;
350
351                 break;
352             }
353         }
354     }
355
356     return $box;
357
358 }
359

```

```

361 public static function filtr_sezon($sezoni,$count)
362 {
363
364     $box = [];
365
366     foreach($sezoni as $k => $v)
367     {
368         foreach($count as $k1 => $v1)
369         {
370
371             $pos = strpos($v1['sezon'],$v);
372
373             if($pos !== false)
374             {
375                 $box[] = $v;
376
377                 break;
378             }
379         }
380     }
381
382     return $box;
383
384 }
385
386 }
387
388 }
389

```

## 2.4.4 Контроллер текстовой страницы – Cont.php

Контроллер Cont.php, реализует веб-страницы с текстовым содержанием (см. стр. [23](#) и [24](#)). Использует файлы вида – cont.php.

**Листинг 6.** Контроллер текстовой страницы - Cont.php.

```
1  <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3  class Controller_Cont extends CMS_Template {    //Template
```

```
4  // class Controller_Cont extends CMS_Controller {    //Controller
```

```
5
```

```
6
```

```
7      public function action_index()
```

```
8      {
```

```
9
```

```
10         $urlsite = URL::site();
```

```
11
```

```
12
```

```
13         $crumbs = CMS::crumbs();
```

```
14
```

```
15         $data = CMS::instance()->get_data();
```

```
16
```

```
17         $this->template->pid = CMS::$pid;
```

```
18         $this->template->title = CMS::$ptitle;
```

```
19         $this->template->pkeys = CMS::$pkeys;
```

```
20         $this->template->pdesc = CMS::$pdesc;
```

```
21         $this->template->content = Twig::factory('cont')
```

```
22             ->set('urlsite', $urlsite)
```

```
23             ->set('crumbs', $crumbs)
```

```
24             ->set('data', $data)
```

```
25             ->set('title', CMS::$pname);
```

```
26     }
```

```
27 }
```

```
28 }
```

## 2.4.5 Контроллер новостей – Novosti.php

Контроллер Novosti.php, реализует страницу списка новостей и одной новости (см. стр. [25](#) и [26](#)). Использует файлы вида novosti.php и novost.php.

**Листинг 7.** Контроллер новостей – Novosti.php

```
1  <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3  class Controller_Novosti extends CMS_Template { //Template
```

```
4  // class Controller_Novosti extends CMS_Controller { //Controller
```

```
5
```

```
6
```

```
7      public function action_index()
```

```
8      {
```

```
9
```

```
10         $urlsite = URL::site();
```

```
11
```

```
12         $url = $this->request->param('param');
```

```
13
```

```
14         $crumbs = CMS::crumbs();
```

```
15
```

```
16         ////////////Одна новость
```

```
17         if($url)
```

```
18         {
```

```
19
```

```
20             $new = CMS::instance()->where('url','=',$url)->get_data();
```

```
21
```

```
22             if(!isset($new[0]))
```

```
23                 throw HTTP_Exception::factory(404,"Page not find");
```

```
24
```

```
25             $this->template->pid = CMS::$parid;
```

```
26             $this->template->title = $new[0]['ptitle'];
```

```
27             $this->template->pkeys = $new[0]['pkeys'];
```

```
28             $this->template->pdesc = $new[0]['pdesc'];
```

```
29
```

```
30             $crumbs[['name']] = $new[0]['title'];
```

```
31
```

```
32
```

```
33             $this->template->content = Twig::factory('novost')
```

```
34                 ->set('urlsite',$urlsite)
```

```
35                 ->set('crumbs',$crumbs)
```

```
36                 ->set('data',$new)
```

```
37                 ->set('title',CMS::$pname);
```

```
38
```

```
39
```

```
40
```

Продолжение листинга на след странице.

```

42
43 }
44 else//////////////////////////////////////// Списание новостей
45 {
46
47     $razdel = $this->request->query('razdel');
48     $page = $this->request->query('page');
49
50     //////////////////////////список разделов
51     $razdeli = CMS::instance()->values_field('razdel');
52
53     //////////////////////////инициализация пагинатора
54     $curr_page = 1;
55     $offset = 0;
56     $limit = 6;
57
58     if(preg_match("/^[\\d]{1,3}$/", $page))
59     {
60         $offset = $limit * ($page-1);
61
62         $curr_page = $page;
63
64     }
65     //////////////////////////////////
66
67     $count = [];
68
69     $curr_razdel = 0;
70

```

```

72
73 if(in_array($razdel,$razdeli))
74 {
75     $news = DB::select('title','date','foto','url','desc')
76         ->from(CMS::$names[0])
77         ->where('visible','=',1)
78         ->where('idpath','=',CMS::$pid)
79         ->where('razdel','like',"%.{$razdel}%.")
80         ->limit($limit)->offset($offset)
81         ->order_by('sort')->execute()->as_array();
82
83     //////////////////////////для подсчета страниц
84     $count = DB::select('id')->from(CMS::$names[0])
85         ->where('visible','=',1)
86         ->where('idpath','=',CMS::$pid)
87         ->where('razdel','like', "%".{$razdel}."%")
88         ->execute()->count();
89     $curr_razdel = $razdel;

```

Продолжение листинга на след странице.

```

92     }
93     else
94     {
95
96         $db = DB::select('title','date','foto','url','desc')
97                     ->from(CMS::$tnames[0])
98                     ->where('visible','=',1)
99                     ->where('idpath','=',CMS::$pid);
100
101         if(!empty($razdeli))
102         {
103             $db->where('razdel','like',"%.{$razdeli[0]}.%");
104             $curr_razdel = $razdeli[0];
105         }
106
107         $news = $db->limit($limit)->offset($offset)
108                     ->order_by('sort')
109                     ->execute()->as_array();
110
111         ////////////////////////////////////////////для подсчета страниц
112         $count = DB::select('id')->from(CMS::$tnames[0])
113                     ->where('visible','=',1)
114                     ->where('idpath','=',CMS::$pid)
115                     ->execute()->count();
116
117     }
118

```

```

119     ////////////////////////////////////////////подсчитаем сколько страниц
120     $pages = ceil($count/$limit);
121
122     ////////////////////////////////// делим по три штуки
123     $box = [];
124     $i = 0;
125     $k1 = 0;
126
127     foreach($news as $k => $v)
128     {
129         if($i%3 == 0) $k1++;
130         //////////////////////////////////изменяем дату
131         $news[$k]['date'] = self::parse_date($news[$k]['date']);
132
133         $box[$k1][$k] = $news[$k];
134
135         $i++;
136     }
137     //////////////////////////////////

```

Продолжение листинга на след странице.

```

138
139     $this->template->pid = CMS::$pid;
140     $this->template->title = CMS::$ptitle;
141     $this->template->pkeys = CMS::$pkeys;
142     $this->template->pdesc = CMS::$pdesc;
143     $this->template->content = Twig::factory('novosti')
144         ->set('urlsite', $urlsite)
145         ->set('uri', CMS::$uri)
146         ->set('razdeli', $razdeli)
147         ->set('news', $box)
148         ->set('pages', $pages)
149         ->set('curr_razdel', $curr_razdel)
150         ->set('curr_page', $curr_page)
151         ->set('title', CMS::$pname);
152
153     }
154
155 }
156

```

```

158     public static function parse_date($newdate = null)
159     {
160
161         $months = array('01' => 'Января',
162             '02' => 'Февраля',
163             '03' => 'Марта',
164             '04' => 'Апреля',
165             '05' => 'Мая',
166             '06' => 'Июня',
167             '07' => 'Июля',
168             '08' => 'Августа',
169             '09' => 'Сентября',
170             '10' => 'Октября',
171             '11' => 'Ноября',
172             '12' => 'Декабря');
173
174         ///////////////парсим дату с y-m-d на d month y
175         $date = explode('-', $newdate);
176
177         $newdate = $date[2].'. ' . ($months[$date[1]]).'. ' . $date[0];
178         ///////////////
179
180         return $newdate;
181
182     }
183
184 }

```



## 2.4.6 Контроллер страницы контактов – Contacts.php

Контроллер Contacts.php, реализует страницу контактов (см. стр. [27](#)).  
Использует файлы вида – contacts.php.

**Листинг 8.** Контроллер страницы контактов – Contacts.php

```
1 <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3 class Controller_Contacts extends CMS_Template { //Template
```

```
4 // class Controller_Contacts extends CMS_Controller { //Controller
```

```
5
```

```
6
```

```
7     public function action_index()
```

```
8     {
```

```
9
```

```
10         $urlsite = URL::site();
```

```
11
```

```
12         $crumbs = CMS::crumbs();
```

```
13
```

```
14         $data = self::group_by('title',CMS::instance()->get_data());
```

```
15
```

```
16         $this->template->pid = CMS::$pid;
```

```
17         $this->template->title = CMS::$ptitle;
```

```
18         $this->template->pkeys = CMS::$pkeys;
```

```
19         $this->template->pdesc = CMS::$pdesc;
```

```
20         $this->template->content = Twig::factory('contacts')
```

```
21             ->set('urlsite',$urlsite)
```

```
22             ->set('crumbs',$crumbs)
```

```
23             ->set('data',$data)
```

```
24             ->set('title',CMS::$pname);
```

```
25     }
```

```
27     public static function group_by($key,$data)
```

```
28     {
```

```
29         $box = [];
```

```
30         $buff = $data;
```

```
31         //////////////Группируем по городам
```

```
32         foreach($data as $k => $v)
```

```
33         {
```

```
34             foreach($buff as $k1 => $v1)
```

```
35             {
```

```
36                 if($v[$key] == $v1[$key])
```

```
37                 {
```

```
38                     $box[$v[$key]][] = $buff[$k1];
```

Продолжение листинга на след странице.

```

39
40         unset($buff[$k1]);
41     }
42 }
43
44
45     return $box;
46
47 }
48

```

## 2.4.7 Контроллер для работы с пользователями – Users.php

Контроллер Users.php, это многофункциональный контроллер для работы с пользователями. Он обрабатывает данные с форм – регистрации пользователя (стр. 28), изменения личных данных пользователя (стр. 32) и вход в личный кабинет (см. стр. 34, Рис.17.1). А так же страница предоставляет страницу личного кабинета. Контроллер использует файлы вида – users\_reg.php, change\_user.php и users.php.

### Листинг 9. Контроллер для работы с пользователями – Users.php

```

1  <?php defined('SYSPATH') or die('No direct script access.');
```

---

```

2
3  class Controller_Users extends CMS_Template {    //Template
4      // class Controller_Users extends CMS_Controller {    //Controller
5
6
7      public function action_index()
8      {
9          //////////////////////////////////////////AJAX
10         if(Request::initial()->is_ajax())////////данные с форм
11         {
12             $post = $this->request->post();
13
14             if(isset($post['reg']))////регистрация пользователя
15                 $this->registration($post);
16             if(isset($post['change']))////изменение пользователя
17                 $this->change_data($post);
18

```

Продолжение листинга на след странице.



```

18
19 ///////////////////////////////////////////////////проверка есть ли такое имя в базе
20 if(isset($post['check_name']))
21 {
22     if(self::check_name($post['check_name']))
23         print('exist');
24     else
25         print('no_exist');
26     exit;
27 }
28
29 ///////////////////////////////////////////////////вход пользователя
30 if(isset($post['in']))
31     $this->user_in($post);
32
33 }
34 ///////////////////////////////////////////////////END AJAX
35
36 $urlsite = URL::site();
37
38 $crumbs = CMS::crumbs();
39
40 $is_live = CMS::user_is_live();
41 ///////////////////////////////////////////////////флаг регистрации
42 $reg = $this->request->query('reg');
43 ///////////////////////////////////////////////////флаг выхода
44 $out = $this->request->query('out');
45
46 ///////////////////////////////////////////////////регистрация
47 if($reg or !$is_live)
48 {
49
50     $crumbs[(count($crumbs)-1)]['name'] = 'Регистрация';
51
52     $this->template->content = Twig::factory('users_reg')
53         ->set('urlsite', $urlsite)
54         ->set('uri', CMS::$uri)
55         ->set('crumbs', $crumbs);
56
57 }
58 elseif($out) ///////////////////////////////////////////////////выход
59 {
60     CMS::user_out();
61     Session::instance()->delete('korzina_tovari');
62     Session::instance()->delete('korzina_result');
63
64     return HTTP::redirect('/');
65 }

```

Продолжение листинга на след странице.

```

66     {
67
68         $crumbs[(count($crumbs)-1)] = ['name' => 'Мой аккаунт'
69                                         , 'url' => 'my-account'];
70         $ch = $this->request->query('change');
71
72         if($ch)
73         {
74             $crumbs[['name']] = 'Изменить личные данные';
75
76             $data = CMS::instance()
77                     ->where('name', '=', $is_live['name'])
78                     ->get_data();
79
80             $this->template->content = Twig::factory('change_user')
81                                     ->set('urlsite', $urlsite)
82                                     ->set('crumbs', $crumbs)
83                                     ->set('data', $data)
84                                     ->set('uri', CMS::$uri)
85                                     ->set('title', CMS::$pname);
86
87         }
88         else
89         {
90             $this->template->content = Twig::factory('users')
91                                     ->set('urlsite', $urlsite)
92                                     ->set('crumbs', $crumbs)
93                                     ->set('uri', CMS::$uri)
94                                     ->set('title', CMS::$pname);
95         }
96     }
97 }
98

```

```

101     ///регистрация пользователя
102     public function registration ($post)
103     {
104         $qap = Session::instance()->get_once('qaptcha_key');
105
106         $validation = Validation::factory($post)
107             ->rule('fname', 'not_empty')
108             ->rule('name', 'not_empty')
109             ->rule('addr', 'not_empty')
110             ->rule('tel', 'Valid::phone')
111             ->rule('email', 'Valid::email')
112             ->rule('pass1', 'not_empty')
113             ->rule('pass2', 'matches',
114                 array(':validation', 'pass2', 'pass1'));
115
116         if($qap and isset($_POST[$qap])
117             and $validation->check()
118             and !CMS::user_name_exist($post['email']))

```

Продолжение листинга на след странице.

```

119 {
120     $id = CMS::user_reg(['name'=>$post['email']
121                        , 'pass'=>$post['pass1']]);
122     if($id > 0)
123     {
124         $update = [
125             'fname'=> $post['fname'], 'rname'=> $post['name'],
126             'name'=> $post['email'], 'addr'=> $post['addr'],
127             'tel'=> $post['tel']
128         ];
129
130         CMS::instance()->update()->set($update)
131             ->where('id', '=', $id)->execute();
132
133         $tovari = Session::instance()->get('korzina_tovari');
134
135         if($tovari)
136             $this->insert_tov($tovari, $post['email']);
137
138         print('no_error');exit;
139     }
140     print('error');exit;
141 }
142 else
143 {
144     print('error');exit;
145 }
146 }

```

```

148     ///изменение данных пользователя
149     public function change_data ($post)
150     {
151
152         $is_live = CMS::user_is_live();
153
154         $validation = Validation::factory($post)
155             ->rule('fname', 'not_empty')
156             ->rule('name', 'not_empty')
157             ->rule('addr', 'not_empty')
158             ->rule('old_pass', 'not_empty')
159             ->rule('pass1', 'not_empty')
160             ->rule('pass2', 'matches'
161                 , array(':validation', 'pass2', 'pass1'));
162
163         if($validation->check()
164             and $is_live
165             and CMS::user_exist(['name' => $is_live['name']
166                                , 'pass' => $post['old_pass']]))
167         {
168
169             $old_data = CMS::instance()->where('name', '=', $is_live['name'])
170                 ->get_data();
171

```

Продолжение листинга на след странице.

```

172 DB::delete('users')->where('name', '=', $is_live['name'])->execute();
173
174 $id = CMS::user_reg(['name'=>$is_live['name'], 'pass'=>$post['pass1']]);
175
176 if($id > 0)
177 {
178
179     $update = [
180         'fname'=> $post['fname'],
181         'rname'=> $post['name'],
182         'name'=> $is_live['name'],
183         'addr'=> $post['addr'],
184         'tel'=> $old_data[0]['tel']
185     ];
186
187     CMS::instance()->update()->set($update)
188         ->where('id', '=', $id)->execute();
189
190     print('no_error');exit;
191
192 }
193
194 print('error');exit;
195
196 }
197 else
198 {
199     print('bad_pass');exit;
200
201 }
202

```

```

205 public static function check_name($name)
206 {
207
208     if(Valid::email($name) and CMS::user_name_exist($name))
209     {
210         return true;
211     }
212     else
213     {
214         return false;
215     }
216
217 }
218
219
220 public function user_in($post)
221 {
222
223     if(isset($post['email'])
224         and isset($post['pass'])
225         and Valid::email($post['email']))
226     {
227         $data['name'] = $post['email'];
228         $data['pass'] = $post['pass'];

```

Продолжение листинга на след странице.

```

230         if(CMS::user_in($data))
231         {
232             $this->sync_tovari($post);
233
234             print('welcome');exit;
235         }
236     }
237
238     print('bad');exit;
239 }
240
241 public function sync_tovari ($post)
242 {
243
244
245     $tovari = Session::instance()->get('korzina_tovari');
246
247     $korzina = DB::select('artikul','imgs','title','zakazano','cena'
248                         , 'summ','email','ostatok')
249                 ->from('korzina')
250                 ->where('email','=',$post['email'])
251                 ->execute()->as_array();
252

```

```

253     /////если пользователь выбрал товары и до этого не авторизовывался.
254     /////то синхронизируем товары
255     if($tovari and $korzina)
256     {
257         $up_flag = false;
258
259         $db_up = DB::update('korzina');
260
261         foreach($korzina as $k1 => $v1)
262         {
263             foreach($tovari as $k => $v)
264             {
265                 /////если в таблице уже были товары,
266                 /////которые пользователь выбрал,
267                 /////заменяем количество старого выбора на новый
268                 if($v1['artikul'] == $v['artikul'])
269                 {
270                     $updata = [
271                         'zakazano' => $v['zakazano'],
272                         'summ' => ($v['cena'] * $v['zakazano']),
273                     ];
274
275                     $db_up->set($updata)->where('email','=',$post['email'])
276                                     ->where('artikul','=',$v['artikul']);
277
278                     $up_flag = true;
279
280                     unset($tovari[$k]);
281                 }
282             }
283         }
284     }

```

Продолжение листинга на след странице.

```

287         if($up_flag) $db_up->execute();
288
289
290         ///если товара еще нет в таблице, то добавляем
291         $this->insert_tov($tovari,$post['email']);
292         //////////////////////////////////////////обновленная корзина
293         $korzina = DB::select('artikul','imgs','title','zakazano'
294                               , 'cena','summ','email','ostatok')
295                               ->from('korzina')
296                               ->where('email','=',$post['email'])
297                               ->execute()->as_array();
298
299         //////////////////////////////////////////обновляем сессию товаров
300         $this->update_sess($korzina);
301
302     }
303     elseif($tovari and empty($korzina))
304     {
305         $this->insert_tov($tovari,$post['email']);
306     }
307     elseif(!$tovari and $korzina)
308     {
309         //////////////////////////////////////////обновляем сессию товаров
310         $this->update_sess($korzina);
311     }
312
313     return true;
314
315 }

```

```

317 public function insert_tov ($tovari,$email)
318 {
319     if(count($tovari) > 0)
320     {
321
322         foreach($tovari as $k => $v)
323         {
324
325             $keys = ['artikul','imgs','title','zakazano'
326                    , 'cena','summ','email','ostatok','idpath'];
327
328             $values = [$v['artikul'],$v['imgs'],$v['title']
329                      , $v['zakazano'],$v['cena']
330                      , ($v['cena'] * $v['zakazano'])
331                      , $email,$v['ostatok'],26];
332
333             DB::insert('korzina', $keys)->values($values)->execute();
334         }
335
336     }
337
338     return true;
339 }

```

Продолжение листинга на след странице.

```

341     public function update_sess($tovari)
342     {
343
344         $total = 0;
345         $summ = 0;
346
347         $sess_tov = [];
348
349         foreach($tovari as $k => $v)
350         {
351             $summ = $summ+($v['cena'] * $v['zakazano']);
352             $total = $total+$v['zakazano'];
353         }
354
355         $result['summ'] = $summ;
356         $result['total'] = $total;
357
358         Session::instance()->set('korzina_tovari',$tovari);
359         Session::instance()->set('korzina_result',$result);
360
361     }
362
363 }

```

## 2.4.8 Контроллер обработки заказов – Zakazi.php

Контроллер Zakazi.php, предназначен для оформления заказов (купить в один клик см. [Рис.17.2](#)) пользователей, а так же, предоставляет список заказов в личном кабинете и подробную информацию каждого заказа (см. стр. [33](#)). Использует файлы вида – zakaz.php, zakazi.php и rez\_zakaz.php.

### Листинг 10. Контроллер обработки заказов – Zakazi.php

```

1  <?php defined('SYSPATH') or die('No direct script access.');
```

---

```

2
3  class Controller_Zakazi extends CMS_Template {    //Template
4      // class Controller_Zakazi extends CMS_Controller {    //Controller
5
6
7      public function action_index()
8      {
9
10         $tovari = Session::instance()->get('korzina_tovari');
11         $result = Session::instance()->get('korzina_result');
12         $is_live = CMS::user_is_live();

```

Продолжение листинга на след странице.



```

14 //////////////////////////////////////////////////оформление заказа
15 if($this->request->is_ajax() and $tovari and $result)
16 {
17
18     $post = $this->request->post();
19
20     //////////////////////////////////////////////////если пользователь авторизован,
21     //////////////////////////////////////////////////оформляем заказ и чистим корзину в базе
22     if($is_live)
23     {
24
25         $valid = Validation::factory($post)
26             ->rule('fname','preg_replace',
27                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
28             ->rule('name','preg_replace',
29                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
30             ->rule('addr','preg_replace',
31                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
32             ->rule('dostavka','preg_replace',
33                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
34             ->rule('tel', 'Valid::phone')
35             ->rule('oplata','preg_replace',
36                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'));
37
38         if($valid->check())
39         {
40
41             $this->set_zakaz($valid->data(), $is_live['name']
42                 , $result, $tovari);
43
44             DB::delete('korzina')->where('email', '=', $is_live['name'])
45                 ->execute();
46
47             Session::instance()->set('zakaz',1);
48
49             echo 'no_error';
50         }
51
52     }
53     //////////////////////////////////////////////////если пользователь не авторизован и оформляет заказ с регистрацией
54     elseif(isset($post['pass1']))
55     {
56
57         $valid = Validation::factory($post)
58             ->rule('fname','preg_replace',
59                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
60             ->rule('name','preg_replace',
61                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
62             ->rule('addr','preg_replace',
63                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
64             ->rule('dostavka','preg_replace',
65                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
66             ->rule('oplata','preg_replace',
67                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ','value'))
68             ->rule('email', 'Valid::email')
69             ->rule('tel', 'Valid::phone')
70             ->rule('pass1', 'not_empty')
71             ->rule('pass2', 'matches',
72                 array(':validation', 'pass2', 'pass1'));

```

Продолжение листинга на след странице.



```

72
73         if($valid->check()
74             and !CMS::user_name_exist($post['email'], null,'users', 14))
75         {
76             ////////////////регистрируем
77             $id = CMS::user_reg(['name'=>$post['email']
78                                 , 'pass'=>$post['pass1']], 'users', 14);
79
80             if($id > 0)
81             {
82                 ////////////////обновляем инфу только
83                 ////////////////что созданного пользователя
84                 $update = [
85                     'fname'=> $post['fname'],
86                     'rname'=> $post['name'],
87                     'name'=> $post['email'],
88                     'addr'=> $post['addr'],
89                     'tel'=> $post['tel']
90                 ];
91
92                 DB::update('users')->set($update)
93                     ->where('id', '=', $id)->execute();
94                 ////////////////
95
96                 $this->set_zakaz($valid->data()
97                                 , $post['email']
98                                 , $result, $tovari);
99
100                 Session::instance()->set('zakaz',1);
101
102                 echo 'no_error';
103             }
104         }

```

```

107     }///если пользователь просто оформляет заказ без регистрации
108     else
109     {
110         $valid = Validation::factory($post)
111             ->rule('fname','preg_replace',
112                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ', ':value'))
113             ->rule('name','preg_replace',
114                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ', ':value'))
115             ->rule('addr','preg_replace',
116                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ', ':value'))
117             ->rule('dostavka','preg_replace',
118                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ', ':value'))
119             ->rule('oplata','preg_replace',
120                 array("/([^\-a-zA-Za-яА-Я0-9ёЁ])/u", ' ', ':value'))
121             ->rule('tel', 'Valid::phone');
122
123         if($valid->check())
124         {
125             $this->set_zakaz($valid->data(),'', $result, $tovari);
126
127             Session::instance()->set('zakaz',1);
128
129             echo 'no_error';
130         }
131     }

```

Продолжение листинга на след странице.

```

132
133     Session::instance()->delete('korzina_tovari');
134     Session::instance()->delete('korzina_result');
135
136     exit;
137
138     }//////////////////////////купить в один клик
139 elseif($this->request->is_ajax() and $this->request->post('one-click'))
140 {
141     $post = $this->request->post();
142
143     $valid = Validation::factory($post)
144         ->rule('one-click','not_empty')
145         ->rule('tel','not_empty');
146
147     $qap = Session::instance()->get_once('qaptcha_key');
148
149     if($qap && isset($post[$qap]) and $valid->check())
150     {
151
152         $insert = [
153             'name' =>$post['name'],
154             'info' =>$post['one-click'],
155             'tel' => $post['tel'],
156             'email' => $post['email']
157         ];
158
159         CMS::instance()->insert($insert,CMS::$tnames[1]);
160
161         ////////////////////////////рассылка уведомлений
162         $body = '<h3>Купить в один клик</h3>'
163             . '<p>Имя заказчика: '.$post['name'].'<br>'
164             . 'Email заказчика: '.$post['email'].'<br>'
165             . 'Телефон: '.$post['tel'].'<br></p>'
166             . '<hr>'
167             . $post['one-click'];
168
169         $from = 'Интернет магазин ВИТЯЗЬ';
170
171         $emails = db::select('emails')->from('confpost')
172             ->execute()->cached()
173             ->get('emails');
174
175         $to = explode('|', $emails);
176
177         $subject = "Купить в один клик - Витязь";
178
179         Libmail::send_html ($body, $from, $to, $subject);
180         ////////////////////////////
181         echo 'success'; exit;
182     }
183 }
184

```

Продолжение листинга на след странице.

```

184     }
185     elseif($is_live)//////////посмотреть мои заказы
186     {
187         $urlsite = URL::site();
188
189         $this->template->pid = CMS::$pid;
190
191         $order = $this->request->query('order');//////////заказ
192
193         if(Valid::digit($order))
194         {
195
196             $data = DB::select([DB::expr('DATE_FORMAT(`date`, "%d.%m.%Y")')
197                 , 'date'], 'nomer', 'status', 'itog'
198                 , 'kolto', 'imzak', 'info', 'addrdos'
199                 , 'addrpl', 'dost', 'vards')
200                 ->from('zakazi')
201                 ->where('email', '=', $is_live['name'])
202                 ->where('nomer', '=', $order)
203                 ->where('visible', '=', 1)
204                 ->execute()->as_array();
205
206             if(empty($data)) HTTP::redirect('/orders');
207
208             $dostavka = DB::select()->from('nastrkor')
209                 ->where('type', '=', 'Вариант доставки')
210                 ->where('name', '=', $data[0]['vards'])
211                 ->execute()->as_array();
212
213             $oplata = DB::select()->from('nastrkor')
214                 ->where('type', '=', 'Метод оплаты')
215                 ->where('name', '=', $data[0]['dost'])
216                 ->execute()->as_array();
217
218
219
220             $this->template->content = Twig::factory('zakaz')
221                 ->set('urlsite', $urlsite)
222                 ->set('data', $data[0])
223                 ->set('nomer', $order)
224                 ->set('dostavka', $dostavka)
225                 ->set('oplata', $oplata)
226                 ->set('title', CMS::$pname);
227         }
228         else
229         {
230             $data = DB::select([DB::expr('DATE_FORMAT(`date`, "%d.%m.%Y")')
231                 , 'date'], 'nomer', 'status'
232                 , 'itog', 'kolto', 'imzak')
233                 ->from('zakazi')
234                 ->where('email', '=', $is_live['name'])
235                 ->where('visible', '=', 1)
236                 ->order_by('nomer', 'desc')
237                 ->execute()->as_array();

```

Продолжение листинга на след странице.

```

238
239         $this->template->content = Twig::factory('zakazi')
240                                     ->set('urlsite',$urlsite)
241                                     ->set('data',$data)
242                                     ->set('title',CMS::$pname);
243     }
244
245     }
246     else HTTP::redirect('/my-account');
247
248 }
249
250 public function set_zakaz($post, $email, $result, $tovari)
251 {
252
253     if(!isset($post['oplata']))
254         $post['oplata'] = '';
255
256     $keys = [
257         'email','status', 'imzak',
258         'itog','info','addrdos',
259         'addrpl','dost','vardos',
260         'koltov','coment','idpath',
261         'visible','tel'
262     ];
263
264     /////привязываем к товару url и проверяем наличие товара
265     $tovari = Controller_Korzina::parse_tov($tovari);
266
267     $info = Twig::factory('rez_zakaz')
268             ->set('tovari',$tovari)
269             ->set('host',$_SERVER['SERVER_NAME'])
270             ->set('totalsumm',$result['summ'])->render();
271
272     $koment = isset($post['koment'])?$post['koment']:'';
273
274     $values = [
275         $email,'Не подтвержден',$post['name'],
276         $result['summ'],$info,$post['addr'],
277         $post['addr'],$post['oplata'],$post['dostavka'],
278         $result['total'],$koment,CMS::$pid,1,
279         $post['tel']
280     ];
281
282     $id = DB::insert('zakazi', $keys)
283           ->values($values)->execute();
284
285     DB::update('zakazi')->set(['nomer' => $id[0]])
286           ->where('id', '=', $id[0])
287           ->execute();

```

Продолжение листинга на след странице.

```

287
288 ////////////////рассылка уведомлений
289 $body = '<h3>Оформление заказа</h3>'
290         .'<p>Имя заказчика: '.$post['name'].'<br>'
291         .'    Email заказчика: '.$email.'<br>'
292         .'    Телефон: '.$post['tel'].'<br>'
293         .'    Итоговая сумма: '.$result['summ'].'<br>'
294         .'    Адрес доставки: '.$post['addr'].'<br>'
295         .'    Метод оплаты: '.$post['oplata'].'<br>'
296         .'    Вариант доставки: '.$post['dostavka'].'<br>'
297         .'    Количество товаров: '.$result['total'].'<br>'
298         .'    Комент покупателя: '.$koment.'<br>'
299         .'    Номер заказа: '.$sid[0].'/p>'
300         .'<hr>'
301         .$info;
302 $from = 'Интернет магазин ВИТЯЗЬ';
303
304
305 $emails = db::select('emails')->from('confpost')
306         ->execute()->cached()->get('emails');
307
308 $to = explode('|', $emails);
309
310
311 $subject = "Оформление заказа - Витязь";
312
313 Libmail::send_html ($body, $from, $to, $subject);
314 ////////////////
315
316 return true;
317
318 }

```

## 2.4.9 Контроллер обработок заявок с сайта – Zakzvon.php

Контроллер Zakzvon.php, собирает заявки с сайта (заказать звонок см. стр. [34 Рис.17](#)), сохраняет их в базе и рассылает на электронные адреса менеджеров. Контроллер не использует файлы вида.

**Листинг 11.** Контроллер обработки заявок – Zakzvon.php

```
1  <?php defined('SYSPATH') or die('No direct script access.');
```

```
2
```

```
3  // class Controller_Zakzvon extends CMS_Template { //Template
```

```
4  class Controller_Zakzvon extends CMS_Controller { //Controller
```

```
5
```

```
6
```

```
7      public function action_index()
```

```
8      {
```

```
9
```

```
10         $urlsite = URL::site();
```

```
11
```

```
12         if(Request::initial()->is_ajax())
```

```
13         {
```

```
14             $qap = Session::instance()->get_once('qaptcha_key');
```

```
15
```

```
16             if($qap && isset($_POST[$qap]))
```

```
17             {
```

```
18
```

```
19                 if($this->request->post())
```

```
20                 {
```

```
21
```

```
22                     $insert['name'] = $this->request->post('name');
```

```
23                     $insert['tel'] = $this->request->post('tel');
```

```
24
```

```
25                     $insert = Arr::map('strip_tags', $insert);
```

```
26
```

```
27                     CMS::instance()->insert($insert);
```

```
28
```

```
29                     ////////////////рассылка уведомлений
```

```
30                     $body = '<h3>Заказ звонка</h3>'
```

```
31                         . '<p>Имя: '.$insert['name'].'<br>'
```

```
32                         . '    Телефон: '.$insert['tel'].'</p>';
```

```
33
```

```
34                     $from = 'Интернет магазин ВИТЯЗЬ';
```

```
35
```

```
36                     $emails = db::select('emails')->from('confpost')
```

```
37                         ->execute()->cached()->get('emails');
```

```
38
```

Продолжение листинга на след странице.

```

38
39         $to = explode('|', $emails);
40
41         $subject = "Пришла заявка из Витязя";
42
43         Libmail::send_html ($body, $from, $to, $subject);
44         //////////////////////////////////////
45
46     }
47
48     echo 'success';
49     exit;
50 }
51 else throw HTTP_Exception::factory(404, "Page not visible");
52 }
53 else throw HTTP_Exception::factory(404, "Page not visible");
54
55 }
56
57 }

```

## 2.4.10 Контроллер корзины – Korzina.php

Контроллер Korzina.php, ведет учет товаров сложенных в корзину, считает сумму и количество товаров, синхронизирует сложенные товары с сессией пользователя и с базой данных (см. стр. [29](#)). Контроллер использует файл вида – korzina.php.

### Листинг 12. Контроллер корзины– Korzina.php

```

1  <?php defined('SYSPATH') or die('No direct script access.');
```

---

```

2
3  class Controller_Korzina extends CMS_Template {    //Template
4      // class Controller_Korzina extends CMS_Controller {    //Controller
5
6
7      public function action_index()
8      {
9
10         $urlsite = URL::site();
11

```

Продолжение листинга на след странице.



```

11
12 //////////////////////////////////////////////////AJAX
13 if(Request::initial()->is_ajax())/////////////////////////////////
14 {
15     $post = $this->request->post();
16
17     if(isset($post['art']))////пополнение корзины
18         $this->set_tovar($post['art']);
19     elseif(isset($post['del']))////удаление с корзины
20         $this->del_tovar($post['del']);
21     elseif(isset($post['min']))////минус один с корзины
22         $this->min_tovar($post['min']);
23     elseif(isset($post['clear']))////очистка корзины
24         $this->clear_tovar();
25 }
26 //////////////////////////////////////////////////AJAX
27
28
29 $tovari = Session::instance()->get('korzina_tovari');
30 $result = Session::instance()->get('korzina_result');
31
32
33 if($tovari and $result)
34 {
35     if($tovari)
36     {
37         //привязываем к товару урл и проверяем наличие товара
38         $tovari = self::parse_tov($tovari);
39     }
40
41
42     $conf = CMS::group_by('type',CMS::instance()
43         ->from(CMS::$tnames[1])->get_data());

```

```

45     $this->template->pid = CMS::$pid;
46     $this->template->title = CMS::$ptitle;
47     $this->template->pkeys = CMS::$pkeys;
48     $this->template->pdesc = CMS::$pdesc;
49
50     $crumbs = CMS::crumbs();
51
52     $is_live = CMS::user_is_live();
53
54     if($is_live)
55     {
56
57         $is_live = DB::select('fname','rname','addr','name','tel')
58             ->from('users')
59             ->where('name','=',$is_live['name'])
60             ->execute()->as_array();
61
62     }

```

Продолжение листинга на след странице.



```

63
64         $this->template->content = Twig::factory('korzina')
65                                     ->set('urlsite', $urlsite)
66                                     ->set('crumbs', $crumbs)
67                                     ->set('tovari', $tovari)
68                                     ->set('result', $result)
69                                     ->set('is_live', $is_live)
70                                     ->set('conf', $conf)
71                                     ->set('title', CMS::$pname);
72
73     }else HTTP::redirect('/');
74
75 }
76
77 public static function parse_tov($tovari)
78 {
79
80     $sub = DB::select('idpath','artikul',['url', 'turl'])
81             ->from('tovari');
82
83     foreach($tovari as $k => $v)
84     {
85         $sub->or_where('artikul','=', $v['artikul']);
86
87     }
88
89     $query = DB::select('a_paths.url','tovari.artikul','tovari.turl')
90             ->from('a_paths')
91             ->join(array($sub, 'tovari'), 'INNER')
92             ->on('a_paths.id', '=', 'tovari.idpath')
93             ->execute()->as_array();
94
95     if(!empty($query))
96     {
97         foreach($tovari as $k => $v)
98         {
99             foreach($query as $k1 => $v1)
100             {
101                 if($v['artikul'] == $v1['artikul'])
102                 {
103                     $tovari[$k]['url'] = $v1['url'];
104                     $tovari[$k]['turl'] = $v1['turl'];
105                 }
106             }
107         }
108     }
109
110     return $tovari;
111
112 }

```

Продолжение листинга на след странице.

```

113
114
115 public function set_tovar($art)
116 {
117
118     $art = preg_replace("/([^\a-zA-Za-яA-Я0-9ёЁ\-\(\)\.\;])/u", " ", $art);
119
120     $stovar = DB::select('title','imgs','cena','artikul')
121         ->from('tovari')->where('artikul','='.$art)|
122         ->execute()->as_array();
123
124     if(!empty($stovar))
125         $stovar[0]['sklad'] = DB::select('ostatok','razmer')
126             ->from('sklad')->where('visible','=1')
127             ->where('artikul','='.$stovar[0]['artikul'])
128             ->execute()->as_array();
129
130     $razmer = $this->razmer_is_exist($this->request->post('t-razmer'));
131     $variac = $this->variac_is_exist($this->request->post('t-variac'));
132
133     if(!empty($stovar) and !empty($stovar[0]['sklad']))
134     {
135
136         $stovari = Session::instance()->get('korzina_tovari');
137
138         $total = 0;
139         $summ = 0;
140
141         if($stovari)
142         {
143
144             foreach($stovari as $k => $v)
145             {
146
147                 if($razmer)
148                 { ///если товар в корзине и имеется еще на складе
149                     if($v['artikul'] == $art and $v['razmer'] == $razmer
150                         and $v['ostatok'] > $v['zakazano'])
151                     {
152                         $stovari[$k]['zakazano']++;
153                         $result['status'] = 'added';
154                     }///если товар в корзине и закончилось количество на складе
155                     elseif($v['artikul'] == $art and $v['razmer'] == $razmer
156                         and $v['ostatok'] == $v['zakazano'])
157                     {
158                         $result['status'] = 'ended';
159                     }
160                 }
161                 else
162                 { /////если товар в корзине и имеется еще на складе
163                     if($v['artikul'] == $art and $v['ostatok'] > $v['zakazano'])
164                     {
165                         $stovari[$k]['zakazano']++;
166                         $result['status'] = 'added';
167                     }/////если товар в корзине и закончилось количество на складе
168                     elseif($v['artikul'] == $art and $v['ostatok'] == $v['zakazano'])

```

Продолжение листинга на след странице.

```

169         {
170             $result['status'] = 'ended';
171         }
172     }
173
174 }
175
176 $tovari[$k]['summ'] = ($v['cena'] * $tovari[$k]['zakazano']);
177
178 $summ = $summ+$tovari[$k]['summ'];
179 $total = $total+$tovari[$k]['zakazano'];
180
181 }
182
183 }
184
185 $user_live = CMS::user_is_live();
186 /////если товара нет в корзине добавляем в корзину
187 if(!isset($result['status']))
188 {
189     if($razmer)/////если у товара есть вариации размеров
190     {
191         foreach($stovar[0]['sklad'] as $k => $val)
192         {
193             if($val['razmer'] == $razmer)
194                 $stovar[0]['ostatok'] = $val['ostatok'];
195         }
196     }
197
198 }
199
200 else
201     $stovar[0]['ostatok'] = $stovar[0]['sklad'][0]['ostatok'];
202
203 $stovar[0]['variac'] = $variac;
204 $stovar[0]['razmer'] = $razmer;
205
206 $result['status'] = 'added';
207 $stovar[0]['zakazano'] = 1;
208
209 $img = explode(' ', $stovar[0]['imgs']);
210
211 if(strlen($img[0]))
212     $stovar[0]['imgs'] = trim($img[0]);
213 else
214     $stovar[0]['imgs'] = '/no-good-image.png';
215
216 $stovar[0]['summ'] = $stovar[0]['cena'];
217
218 $tovari[] = $stovar[0];
219
220 $summ = $summ+$stovar[0]['cena'];
221 $total++;

```

Продолжение листинга на след странице.

```

221
222     ///если пользователь зарегистрирован
223     if($user_live)
224     {
225         $insert = [
226             'artikul' => $stovar[0]['artikul'],
227             'imgs' => $stovar[0]['imgs'],
228             'title' => $stovar[0]['title'],
229             'zakazano' => 1,
230             'cena' => $stovar[0]['cena'],
231             'summ' => $stovar[0]['cena'],
232             'ostatok' => $stovar[0]['ostatok'],
233             'email' => $user_live['name'],
234             'razmer' => $razmer,
235             'variac' => $variac,
236         ];
237
238         CMS::instance()->insert($insert);
239     }
240
241     }///если произошло событие (если товар в корзине и имеется еще на складе)
242     ///и пользователь авторизован. обновляем инфу в корзине
243     elseif($result['status'] == 'added' and $user_live)
244     {
245         $data = CMS::instance()->update()
246         ->set([
247             'zakazano' => DB::expr('zakazano + 1'),
248             'summ' => DB::expr('summ + '.$stovar[0]['cena']),
249         ])
250         ->where('email','=',$user_live['name'])
251         ->where('artikul','=',$stovar[0]['artikul']);
252
253         if($razmer) $data->where('razmer','=',$razmer);
254
255         $data->execute();
256
257     }
258
259     $result['summ'] = $summ;
260     $result['total'] = $total;
261
262     Session::instance()->set('korzina_tovari',$stovari);
263     Session::instance()->set('korzina_result',$result);
264
265     $result['summ'] = number_format($summ,0,' ',' ');
266
267     echo json_encode($result);
268     exit;
269
270 }
271
272
273
274
275
276
277 public function del_tovar($art)
278 {

```

Продолжение листинга на след странице.

```

279
280     $user_live = CMS::user_is_live();
281
282     $art = preg_replace("/([^\a-zA-Za-яA-Я0-9ёё\-\(\)\.\;])/u", " ", $art);
283
284     $tovari = Session::instance()->get('korzina_tovari');
285     $result = Session::instance()->get('korzina_result');
286
287     $razmer = $this->razmer_is_exist($this->request->post('t-razmer'));
288     $variac = $this->variac_is_exist($this->request->post('t-variac'));
289
290     $flag = false;
291
292     foreach($tovari as $k => $v)
293     {
294         if($art == $v['artikul'] and $result['total'] > 0
295            and $result['summ'] > 0)
296         {
297
298             if($razmer)
299             {
300                 if($v['razmer'] == $razmer)
301                     $flag = true;
302             }
303             else
304             {
305                 $flag = true;
306             }
307
308             if($flag)
309             {
310
311                 $result['total'] = $result['total'] - $v['zakazano'];
312                 $result['summ'] = $result['summ'] - $v['summ'];
313
314                 if($user_live)/////если пользователь зарегистрирован
315                 {
316
317                     $db = DB::delete('korzina')
318                         ->where('email', '=', $user_live['name'])
319                         ->where('artikul', '=', $v['artikul']);
320
321                     if($razmer) $db->where('razmer', '=', $v['razmer']);
322
323                     $db->execute();
324
325                 }
326
327                 unset($tovari[$k]);
328
329                 break;
330
331             }
332
333         }
334
335     }

```

Продолжение листинга на след странице.

```

336
337         if($result['total'] <= 0 and $result['summ'] <= 0)
338         {
339             Session::instance()->delete('korzina_tovari');
340             Session::instance()->delete('korzina_result');
341         }
342         else
343         {
344             Session::instance()->set('korzina_tovari',$tovari);
345             Session::instance()->set('korzina_result',$result);
346         }
347
348         $result['status'] = 'deleted';
349
350         echo json_encode($result);
351         exit;
352
353     }
354
355     public function clear_tovar()
356     {
357
358         $user_live = CMS::user_is_live();
359
360         Session::instance()->delete('korzina_tovari');
361         Session::instance()->delete('korzina_result');
362
363         if(isset($user_live['name']))
364         {
365
366             DB::delete('korzina')->where('email', '=', $user_live['name'])
367                 ->execute();
368
369         }
370
371         exit;
372     }
373
374
375     public function min_tovar($art)
376     {
377
378         $user_live = CMS::user_is_live();
379
380         $art = preg_replace("/([a-zA-Za-яA-Я0-9äË\-\(\)\.\;])/u", " ", $art);
381
382         $tovari = Session::instance()->get('korzina_tovari');
383         $result = Session::instance()->get('korzina_result');
384
385         $razmer = $this->razmer_is_exist($this->request->post('t-razmer'));
386         $variac = $this->variac_is_exist($this->request->post('t-variac'));
387
388         $flag = false;

```

Продолжение листинга на след странице.

```

390         foreach($tovari as $k => $v)
391         {
392             if($art == $v['artikul'] and $v['zakazano'] > 1)
393             {
394                 if($razmer)
395                 {
396                     if($v['razmer'] == $razmer)
397                     {
398                         $flag = true;
399                     }
400                 }
401                 else
402                 {
403                     $flag = true;
404                 }
405             }
406             if($flag)
407             {
408                 $tovari[$k]['summ'] = $tovari[$k]['summ'] - $v['cena'];
409                 $tovari[$k]['zakazano']--;
410                 $result['total']--;
411                 $result['summ'] = $result['summ'] - $v['cena'];
412                 if($user_live)////////если пользователь зарегистрирован
413                 {
414                     $db = CMS::instance()->update()
415                     ->set([
416                         'zakazano' => DB::expr('zakazano - 1'),
417                         'summ' => DB::expr('summ - '.$v['cena']),
418                     ])
419                     ->where('email','=',$user_live['name'])
420                     ->where('artikul','=',$v['artikul']);
421                 }
422             }
423             if($razmer) $db->where('razmer','=',$v['razmer']);
424             $db->execute();
425         }
426         break;
427     }
428 }
429
430
431
432
433
434
435 }
436
437 Session::instance()->set('korzina_tovari',$tovari);
438 Session::instance()->set('korzina_result',$result);
439
440 $result['status'] = 'deleted';
441
442 echo json_encode($result);
443 exit;
444
445
446 }

```

Продолжение листинга на след странице.



```

447
448     public function variac_is_exist($name)
449     {
450         $variac = CMS::values_field('variac');
451
452         return in_array($name,$variac)?$name:false;
453     }
454
455     public function razmer_is_exist($name)
456     {
457         $razm = CMS::values_field('razmer');
458
459         return in_array($name,$razm)?$name:false;
460     }
461
462
463
464 }

```

### 2.4.11 Контроллер отзывов – Otvivi.php

Контроллер Otvivi.php, предназначен для обработки отзывов и рассылки на электронные адреса менеджеров интернет-магазина.

**Листинг 13.** Контроллер отзывов – Otvivi.php

```

1  <?php defined('SYSPATH') or die('No direct script access.');
```

---

```

2
3  class Controller_Otvivi extends CMS_Template {    //Template
4  // class Controller_Otvivi extends CMS_Controller {    //Controller
5
6
7      public function action_index()
8      {
9
10         $urlsite = URL::site();
11
12         if(Request::initial()->is_ajax())
13         {
14             $post = $this->request->post();
15
16             $object = Validation::factory($post);
17

```

Продолжение листинга на след странице.

```

18 $object->rule('name', 'not_empty')
19     ->rule('name', 'max_length', array(':value', '30'))
20     ->rule('otziv', 'not_empty')
21     ->rule('otziv', 'max_length', array(':value', '100'))
22     ->rule('ocenka', 'digit');
23
24 if ($object->check())
25 {
26
27     CMS::instance()->insert([
28         'ocenka' => $post['ocenka'],
29         'otziv' => $post['otziv'],
30         'name' => $post['name'],
31     ]);
32
33     ////////////рассылка уведомлений
34     $body = '<h3>Новый отзыв</h3>'
35         .'<p>Имя заказчика: '.$post['name'].'<br>'
36         .'    Оценка: '.$post['ocenka'].'<br></p>'
37         .'<hr>'
38         .$post['otziv'];
39
40     $from = 'Интернет магазин ВИТЯЗЬ';
41
42     $emails = db::select('emails')->from('confpost')
43         ->execute()->cached()
44         ->get('emails');
45
46     $to = explode('|', $emails);
47
48     $subject = "Отзыв - Витязь";
49
50     Libmail::send_html ($body, $from, $to, $subject);

```

```

52
53     }
54
55     exit;
56 }
57 else
58 {
59     $this->template->pid = CMS::$pid;
60     $this->template->title = CMS::$ptitle;
61     $this->template->pkeys = CMS::$pkeys;
62     $this->template->pdesc = CMS::$pdesc;
63
64     $data = CMS::instance()->get_data();
65
66     foreach($data as $k => $val)
67     {
68         $d = explode(' ', $val['date']);

```

```

69
70         $data[$k]['date'] = CMS::parse_date($d[0]);
71     }
72
73     $this->template->content = Twig::factory('otzivi')
74         ->set('crumbs', CMS::crumbs())
75         ->set('data', $data)
76         ->set('uri', CMS::$uri);
77     }
78 }
79
80 }

```

Это были все контроллеры модулей, с помощью которых функционирует интернет-магазин – «Витязь». Были пропущены два контроллера это – Mainpage.php и Sklad.php. В этих контроллерах нет программного кода, они служат «заглушками» для разделов CMS системы, так как по правилам CMS, нельзя создать раздел, не присвоив к нему модуль. Mainpage.php – служит для раздела CMS – Главная, где данные с главной страницы. Sklad.php – служит для раздела – Склад, где информация о количестве товаров, если на складе нет определенного товара, то в интернет-магазине этот товар будет под статусом – Нет на складе.

## Заключение

Разработка с нуля интернет-магазина, весьма затратный и длительный процесс для заказчика. Ведь для того что бы разработать интернет-магазин, который бы приносил прибыль, необходимо, что бы он отвечал определенным требованиям, таким как – высокая производительность, дружелюбный дизайн, уникальный контент, и многое другое. А для того что бы достичь этих целей необходимо привлечь определенных специалистов, это:

- Менеджер проекта - формирует будущую структуру, визуальное оформление сайта и составляет план предстоящих работ, которые заканчиваются утверждением с заказчиком.
- Дизайнер - формирует визуальное пространство сайта, используя различные элементы, как в целях повышения функциональности ресурса, так и его украшения.
- Верстальщик - преобразует макет дизайна, в веб страницу, при помощи различных элементов языка разметки, стилизации и скриптов.
- Программист – завершает процесс разработки интернет-магазина, используя верстку сайта и серверный язык программирования.
- SEO специалист – Заполняет веб сайт уникальным контентом для продвижения в поисковых системах.

В некоторых случаях, готовый интернет-магазин может быть рациональным выбором, однако в большинстве случаев подобный выбор имеет существенный ряд минусов. Это и сложность настройки под конкретные требования, громоздкость кода, которая будет приводить к замедлению работы сайта, это и, зачастую, проблемы с безопасностью. Таким образом, самостоятельное изучение веб-программирования будет, безусловно, предпочтительным для любого, кто решил заняться созданием своих собственных сайтов.

## Литература

1. Гуров Ф.Н. Продвижение бизнеса в Интернете: все о PR и рекламе в сети – М.: Вершина, 2008. – 136с.
2. Скотт Д.М. Новые правила маркетинга и PR : как использовать социальные сети, блоги, подкасты и вирусный маркетинг для непосредственного контакта с покупателями / пер. с англ. В. Апанасик, Г. Огибин ; ред. А. Членова. 2-е изд. М.: Альпина Паблишерз, 2011. 347 с.
3. Бесчастная Е.В. Интерактивный маркетинг и электронная коммерция: Электронный учебно-методический комплекс для студентов специальности I -26 02 03 Маркетинг. – Мн.: БГУИР, 2007. – 303 с.
4. Калинина А.Э. Интернет-бизнес и электронная коммерция: Учебное пособие. – Волгоград: Изд-во ВолГУ, 2004. – 148 с.
5. Технологии создания систем электронной коммерции: учеб.-метод. пособие / К. А. Юрков, Л. Н. Лядова, А. В. Хлызов, Г. В. Климов; Перм. гос. ун- т. Пермь, 2007. 80 с.
6. Ладонина Л. Книга руководителя интернет-проекта. Готовые маркетинговые решения. – СПб.: ПИТЕР, 2008. – 256 с.
7. Кильдеева Р.А. Составление бизнес-плана предприятия. Теория и практика: Учебное пособие. – Петропавловск-Камчатский: КамчатГТУ, 2005. – 150 с.
8. Репин В.В. Бизнес-процессы компании: построение, анализ, регламентация – М.: РИА «Стандарты и качество», 2007. – 240 с., ил. – (Серия «Деловое совершенство»).
9. Манн И. и др. Интернет-маркетинг на 100%. М., 2011. 240 с.
10. Царев В.В., Кантарович А.А. Электронная коммерция. – СПб: Питер, 2002. – 320 с.

11. Интернет-маркетинг и электронная коммерция: учеб.- метод. пособие / К. В. Новикова, А. С. Старатович, Э. А. Медведева; Перм. гос. нац. исслед. ун-т. – Пермь, 2013. – 78 с.
12. Дик В.В., Лужецкий М.Г., Родионов А.Э., Электронная коммерция. / Московская финансово-промышленная академия. – М., 2005. – 376 с.
13. Шматалюк А., Ферапонтов М., Громов А., Каменова М. Моделирование бизнеса. Методология ARIS – 327 с. 2001.
14. Юрасов А.В. Основы электронной коммерции: учеб. для вузов / Юрасов А.В. – М.: Горячая линия. – Телеком, 2008. – 480 с.
15. Юрасов А.В. Электронная коммерция: Учеб. пособие. – М.: Дело, 2003. – 480 с.
16. Бен Фрей HTML5 и CSS3 Разработка сайтов для любых браузеров и устройств. – 254 с.
17. Веру Л Секреты CSS. Идеальные решения ежедневных задач. — СПб.: Питер, 2016. — 336 с.: ил. — (Серия «Бестселлеры O'Reilly»).
18. Макфарланд Д. Новая большая книга CSS. — СПб.: Питер, 2016. — 720 с.: ил. — (Серия «Бест- селлеры O'Reilly»).
19. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. — СПб.: Питер, 2012. — 304 с.: ил.
20. Колисниченко Д. Н. К60 PHP и MySQL. Разработка Web-приложений. 4-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2013. 560 с.: ил. — (Профессиональное программирование)
21. Завстра, Мэт. PHP: объекты, шаблоны и методики программирования, 4-е изд . Пер. с англ. - М.: ООО "И.Д. Вильямс", 2015. - 576 с.: ил. - Парал . тит. англ .
22. Прохоренок Н . А . HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2010. 912 с.: ил. + Видеокурс (на CD-ROM) — (Профессиональное программирование)

23. Welling L., Thomson L. PHP and MySQL Web Development (4th Edition). Addison-Wesley Professional, 2008 – 1008 p.

24. Мэтью Д. HTML5. Разработка веб-приложений. М.: Рид Групп, 2012 – 320 с.

Интернет источники:

25. <http://php.net/> - официальный сайт языка программирования php.

26. <http://kohanaframework.org/> - официальный сайт php HMVC – фреймворка.

27. <http://kohanaframework.su/> - Русскоязычный сайт, посвященный фреймворку Kohana.

28. [Htmlbook.ru](http://htmlbook.ru) – русскоязычный справочник по html и css.

29. [Ruseller.com](http://ruseller.com) – css учебник и уроки по html и многим другим тематикам, полезным для вебмастера.

30. [Htmldog.com](http://htmldog.com) – оригинальный ресурс с уроками разного уровня сложности.